ITIL Release Management

A Hands-on Guide



Dave Howard

CRC Press
Taylor & Francis Group
AN AUERBACH BOOK

ITIL Release Management

A Hands-on Guide

© 2010 by Taylor and Francis Group, LLC

ITIL Release Management

A Hands-on Guide

Dave Howard



CRC Press
Taylor & Francis Group
Boca Raton London New York

CRC Press is an imprint of the
Taylor & Francis Group, an **informa** business
AN AUERBACH BOOK

© 2010 by Taylor and Francis Group, LLC

CRC Press
Taylor & Francis Group
6000 Broken Sound Parkway NW, Suite 300
Boca Raton, FL 33487-2742

© 2010 by Taylor and Francis Group, LLC
CRC Press is an imprint of Taylor & Francis Group, an Informa business

No claim to original U.S. Government works

Printed in the United States of America on acid-free paper
10 9 8 7 6 5 4 3 2 1

International Standard Book Number: 978-1-4398-1558-8 (Hardback)

This book contains information obtained from authentic and highly regarded sources. Reasonable efforts have been made to publish reliable data and information, but the author and publisher cannot assume responsibility for the validity of all materials or the consequences of their use. The authors and publishers have attempted to trace the copyright holders of all material reproduced in this publication and apologize to copyright holders if permission to publish in this form has not been obtained. If any copyright material has not been acknowledged please write and let us know so we may rectify in any future reprint.

Except as permitted under U.S. Copyright Law, no part of this book may be reprinted, reproduced, transmitted, or utilized in any form by any electronic, mechanical, or other means, now known or hereafter invented, including photocopying, microfilming, and recording, or in any information storage or retrieval system, without written permission from the publishers.

For permission to photocopy or use material electronically from this work, please access www.copyright.com (http://www.copyright.com/) or contact the Copyright Clearance Center, Inc. (CCC), 222 Rosewood Drive, Danvers, MA 01923, 978-750-8400. CCC is a not-for-profit organization that provides licenses and registration for a variety of users. For organizations that have been granted a photocopy license by the CCC, a separate system of payment has been arranged.

Trademark Notice: Product or corporate names may be trademarks or registered trademarks, and are used only for identification and explanation without intent to infringe.

Library of Congress Cataloging-in-Publication Data

Howard, Dave, 1957-
ITIL release management : a hands-on guide / Dave Howard.
p. cm.
Includes bibliographical references and index.
ISBN 978-1-4398-1558-8 (alk. paper)
1. Information technology--Management. 2. Information technology projects--Management. 3. Software support--Management. 4. Configuration management. I. Title.

T58.64H69 2010
004.068--dc22 2009042193

Visit the Taylor & Francis Web site at
http://www.taylorandfrancis.com

and the CRC Press Web site at
http://www.crcpress.com

© 2010 by Taylor and Francis Group, LLC

Contents

Introduction **xi**

1 Overview **1**
What Is Release Management? 1
Information Technology Infrastructure Library (ITIL) Service Management 2
Financial Benefits 3
Release Management and Project Management 4
Organizational Readiness 5
Practical Application 6
Baseline and Metrics 6
Concept Review 7

2 Basic Concepts **9**
Objective and Mission 9
Release Policy and Procedures 12
Guidance 13
Facilitator 13
Governance 13
Release Standards 14
Basic Concepts 14
Release Models 14
Deployment Considerations 16
Release Schedules 17
Naming Conventions 18
Release Procedures 21
Emergency Releases 21
Roles and Responsibilities 22
Release Lifecycle 24
Definitive Media Library 24
Measures and Metrics 25

© 2010 by Taylor and Francis Group, LLC

Release Management Activities....25
Developmental Relationships....26
Operational Considerations/Total Cost of Ownership....29
Concept Review....30

3 Release Management and Project Management: Kissing Cousins....33
Differences....33
Similarities and Complementary Likeness....35
Release Plan and Project Plan....36
Tying It Together....36
Software/System Development Lifecycle (SDLC) vs. the Release Lifecycle (RLC)....39
Alignment Pyramid....41
Concept Review....43

4 Release Management Is Not Just for Software Development....45
Holistic Approach....45
Release Management and the Infrastructure....46
New Environment Request Process....46
NER Process Description....48
Step 3.0 New Environment Request Form (NERF)....49
Step 3.1 NER Review Meeting....50
Step 3.2 Produce the Infrastructure Design Document....50
Step 3.3 Design Approval....51
Step 3.4 Infrastructure Procurement....52
Step 3.5 Create the Configuration Items....52
Step 3.6 Build Environment....53
Step 3.7 Operational Readiness Testing....53
Step 3.8 Implement and Turnover....54
Concept Review....54

5 The Release Lifecycle....57
Benefits....57
Implementation Considerations....58
The Release Lifecycle....59
Artifacts....61
Initiation Phase....62
Step 1.0 Initiation....62
Step 1.1 Release Review....63
Step 1.2 Planning Meeting....64
Step 1.3 Release Plan....65
Step 1.4 Release Initiated....65

© 2010 by Taylor and Francis Group, LLC

Benefits 66
Artifacts and Documents 66
Release Configuration and Build Phase 67
Step 2.0 Business Requirements and Functional Specifications 69
Step 2.1 Develop Application 69
Step 2.2 Code Accepted 70
Step 2.3 Deliver Code to DML 70
Step 2.4 Release Validated 71
Step 2.5 Build and Configure 71
Benefits 72
Artifacts/Documents 72
New Environment Request (NER) Phase 73
Benefits 74
Artifacts/Documents 74
Quality Assurance Phase 76
Step 4.0 Master Test Plan 77
Step 4.1 Release Testing Strategy 77
Step 4.2 Test Execution Plan 78
Step 4.3 Technical Test Execution 79
Step 4.4 User Acceptance Testing 80
Step 4.5 QA Acceptance 81
Benefits 82
Artifacts/Documents 82
Operational Readiness Phase 83
Step 5.0 Operational Readiness Entrance 83
Step 5.1 Final Operational Readiness Testing (ORT) 84
Step 5.2 Final QA Report 85
Step 5.3 Support, Escalation, and Turnover 85
Early Life Support 85
Step 5.4 Master Training Plan 88
Step 5.5 Service Level Agreement 88
Step 5.6 IT Service Continuity 89
Step 5.7 Final Operational Readiness Review 90
Benefits 90
Artifacts/Documents 91
Implementation Phase 92
Step 6.0 Implementation and Back-Out Plan 92
Step 6.1 Change Management 93
Step 6.2 Implement Release 94
Benefits 94
Artifacts/Documents 94
Post-Implementation Phase 95
Step 7.0 Post-Implementation Review/Lessons Learned 95

© 2010 by Taylor and Francis Group, LLC

Step 7.1 Complete Release Record 96
Step 7.2 Measure KPIs and Report 97
Step 7.3 Close Release 98
Benefits 98
Artifacts/Documents 98
Using the RLC 99
Right Sizing 99
Service Management Alignment 101
Change Management 101
Configuration Management 102
Availability and Capacity Management 103
Incident Management/Service Desk 103
Problem Management 104
Service Level Management 104
IT Service Continuity 104
Continual Service Improvement 105
IT Security Management 105
Knowledge Management 105
Procurement Management 105
Concept Review 106

6 Release Measures and Metrics 107
Setting the Baseline 107
Value of Measurement 108
Components of Metrics 110
Measurement and Metrics 113
Concept Review 115

7 Selling Release Management 117
Organizational Readiness 118
Executive Buy-In 118
Middle Management 119
Grassroots 120
Bubble-Up Methodology 121
Return on Investment 122
Release Management and the Business 124
Concept Review 124

Appendix A: Release Policy 127

Appendix B: RACI Matrix 153

Appendix C: Release Deliverables Checklist 159

Appendix D: Master Test Plan 163

© 2010 by Taylor and Francis Group, LLC

Appendix E: New Environment Request Form (NERF).....183
Appendix F: Infrastructure Design Document (IDD) 191
Appendix G: Security Risk Assessment.....201
Appendix H: Functional Specifications.....209
Appendix I: Operational Readiness Testing Manual (ORT).....225
Appendix J: Support and Escalation Turnover Document.....255
Appendix K: Master Training Plan.....271
Appendix L: Service Level Agreement.....277
Appendix M: IT Service Continuity and Recovery Plan.....285
Appendix N: Post-Implementation Review 305

© 2010 by Taylor and Francis Group, LLC

Introduction

There once was a very successful company that relied on a parent company to supply its information technology (IT) capabilities; the company had an IT operation that was really a simple application-development shop. The company, Acme Finance, a mid-sized financial services company, had different IT needs than the parent company, which was a distribution company. It was determined that this model would not allow Acme to achieve its growth and profitability goals and objectives. As a result, the decision was made to separate IT operations from the parent company. This decision meant that Acme needed to become a full-service IT operation and implement processes that would allow it to design, develop, implement, and operate IT services that would meet the needs of Acme's business units. After a review of several frameworks it was decided that Acme would use the Information Technology Infrastructure Library (ITIL) framework as the guiding framework on which to base its service delivery model.

As Acme continued through its ITIL journey it came to the point of looking at its development processes and how it developed and released new or modified services into operation. Referring to the ITIL framework for direction, Acme started looking at the release management chapter in the service support or "blue book." What it found was good high-level theory, but it did not find real practical guidance on how to use the concepts discussed in the book. Acme went searching for white papers, books, and any other guidance on how to implement release management, but to its surprise it really didn't find any tangible guidance; so it was on its own to try to find its way. Acme, like many other companies, learned many lessons and successful concepts through a lot of trial and error and determination, which eventually led to a successful implementation of release management. The purpose of this book is to provide a resource for organizations embarking on the journey of implementing a release management practice. Sharing lessons learned and providing practical process ideas will give the reader a useful resource to successfully implement a release process.

© 2010 by Taylor and Francis Group, LLC

Chapter 1

Overview

There are many facets to the successful implementation of a release management process; they range from understanding the basic concepts to the benefits derived for the business user. Managing cultural aspects such as selling the benefits, gaining executive support, and determining organizational readiness are as important as creating good processes. Trying to implement a process without the consideration and balance of all of these factors will not be successful. With all implementations there are some components that are more important than others; however, without some traces of the critical success components the implementation will not be balanced. This book will discuss the various components and concepts important to a successful implementation and how they relate to each other.

What Is Release Management?

Release management is a process that describes a controlled method of providing consultative guidance, scheduling, and governance of changes to a specific service or product. Taking into consideration the holistic view of the service, release management, if utilized properly, focuses on ensuring the quality of the service from inception to retirement; this is a "total cost of ownership" approach. By taking this holistic view, the service that is being provided maintains a higher level of availability, costs less to maintain, and increases the overall stability and reliability of the service. When services being provided to the business experience increased availability and stability, the business has the ability to increase profitability.

An added benefit of creating a controlled, consistent, and repeatable process that delivers quality releases on time, on budget, and within requirements is improving the image of IT with the business. This is particularly important in today's world

© 2010 by Taylor and Francis Group, LLC

as IT becomes more of a commodity every day. Instead of becoming a commodity, using controlled and planned processes that align with the business strategy, IT moves from being a commodity to becoming a strategic asset.

Information Technology Infrastructure Library (ITIL) Service Management

A book about release management would be incomplete without the mention of Information Technology Infrastructure Library (ITIL) service management and how release management fits within the lifecycle. ITIL service management provides best practice guidance on developing effective service delivery processes. The most recent version of ITIL utilizes a lifecycle approach to provide delivery of services and management of those services. The lifecycle consists of five functions, service strategy, service design, service transition, service operation, and continual service improvement; each function has its distinct role, yet they all relate in some form to the holistic lifecycle. While this book is not about ITIL and familiarity with the framework is not a prerequisite to using the release management concepts discussed in this book, a general understanding of the ITIL lifecycle and how release management fits within the lifecycle will enhance the level of effectiveness. A general review of the ITIL framework (see Figure 1.1) will provide this understanding.

As with any successful effort, having a sound strategy is the keystone to success, and ITIL is no different; the lifecycle revolves around the Service Strategy function.

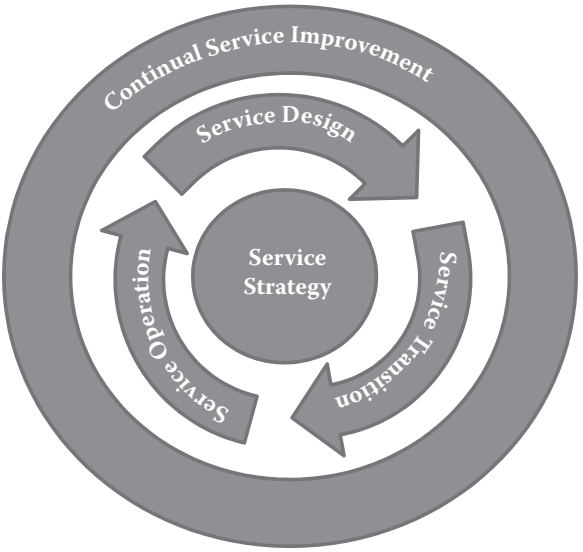


Figure 1.1 The ITIL Service lifecycle. ITIL® is a registered trademark of the Office of Government Commerce (OGC).

© 2010 by Taylor and Francis Group, LLC

During this phase of the lifecycle, an IT organization needs to understand the strategy and objectives of the business and how it can enable them. When IT strategies are aligned to business strategies, IT starts to become a strategic asset and partner to the business. Having an IT strategy that aligns with the business strategy is a key component of executing a successful release management process. Business services are usually created as enablers for the business. When these services are built, implemented, and operated, keeping a strategy focus enables quality delivery. This is a primary goal of release management.

Sustaining services and continual improvement is also part of the ITIL lifecycle. A function within the lifecycle called Continual Service Improvement (CSI) focuses on driving these concepts. The continual service improvement process (CSIP) provides a seven-step process to help define this activity. Another part of improving the delivery of services is using the Deming Cycle or the PDCA approach—plan, do, check, act. If the reader is not familiar with PDCA concepts, it is recommended to consult the ITIL books for a better understanding. Many times the CSI function and processes drive the improvements of the services that are developed and implemented using the release process.

Release and deployment management can be found in the Service Transition function within the ITIL lifecycle. While this seems like a natural fit for a process that focuses on the development and transition of changes to a service, there are activities that are inputs and outputs to other functions within the lifecycle, such as service operations and CSI. Additionally, a successful release process should be aligned with the nucleus of the lifecycle, service strategy. In Chapter 2, the principals and concepts of release and deployment management described within the Service Transition function will be discussed and serve as a baseline for understanding how to practically implement release management.

Financial Benefits

Reducing cost and improving organizational profitability has always been a cornerstone of a successful IT operation. Chief information officers (CIOs) have made their careers on being able to reduce IT costs and improve the performance of their IT operations. Implementing an effective release management process can have a significant impact on the efficiency and the cost of developing, implementing, and delivering services, applications, and infrastructure. One financial benefit that is often overlooked is improved system availability and reduced downtime of IT services realized by utilizing good release practices. The financial benefit gained from increased availability is realized when users are able to utilize the IT services to generate revenue and provide better customer service. When these services are unavailable, users are unable to generate revenue. Another aspect of reduced downtime or incidents is the cost of resources expended to restore service. These expenses are generally realized within the

© 2010 by Taylor and Francis Group, LLC

operational budget and do not add revenue to the bottom line, and therefore should be minimized when possible.

Improved financial transparency can also be achieved through efficient release planning and scheduling. When utilized properly, release schedules and release calendars provide transparency to manage release activity cost, resource cost, and procurement cost. Being able to plan and manage these expenditures over an extended period of time now moves the IT spending into a strategic position for the organization. Release scheduling has other benefits to the business that will be discussed later.

During the release process, assets such as service level agreements, support documentation, and release notes will be generated. These assets are knowledge capital that are used to deliver and operate the service. The cost to produce these assets can be expensive and are a significant part of the release budget. Once the original release has been created and implemented, these release assets now become the baseline from which the next release is generated; this is when the concept of reusability enters the picture. When the next version of the release is under development, project teams are able to refer to these original assets as a starting point rather than having to start at the beginning and reinvent the assets, which saves time and funding. Additionally, some assets, such as support documentation and service level agreements (SLAs) may not need to be updated since they haven't changed. This reusability therefore reduces the release cost and the time to delivery.

These factors are not always recognized as financial benefits to the organization since many times the facilities to realize these savings are not in place, or the organization is not mature enough to measure the benefits. It is therefore important to create a baseline so the financial benefits can be forecast and measured; additional discussion of metrics will be covered in Chapter 6.

Release Management and Project Management

When implementing release management there are day-to-day practical challenges that need to be overcome to ensure success. One of the most common issues is the confusion between release management and project management and understanding the differences. The common cause of this confusion is not aligning the objectives of each and understanding how they relate to each other. While there are similarities between the two, release practices complement project management processes and vice versa, there are also differences between the two. The relationship between release management and project management can be a love–hate relationship or they can live in harmony—it is dependent on how both are positioned. The basic difference is that release management takes into consideration the holistic approach of the entire service, and project management has a specific focus with a beginning and an end. This relationship and how to cultivate a positive partnership is the topic of Chapter 3.

© 2010 by Taylor and Francis Group, LLC

Organizational Readiness

Anytime a new process or an existing process is changed there will always be those who embrace the change and become the biggest advocates and those who have trouble accepting it. While there are steps that can be taken to move the pendulum to the supportive side, there will always be those doubting the validity of the new process. The first hurdle to overcome is the lack of understanding of the goals and objectives of release management. A common mistake that is made is trying to mandate the adoption of the process and its requirements without providing the "whys"—why the new process is being implemented and why the current process is being changed. Another important detail that is overlooked is the "What's In It For Me" or the WIIFM issue. Generally people are much more receptive to changes if they can understand how it benefits them personally. Once resources understand the what's in it for me and the whys, they should have a clear understanding of the goals and objectives. The next obstacle is to ensure that any resource involved with the release process understands release concepts and the value these concepts bring to the organization and the operation of the service.

In the real world we have to realize that no matter how good a job is done trying to explain the goals and objectives—the what's in it for me issue, the whys, and the concepts—there will always be people who are going to be doubters. However, by properly planning and communicating, we can reduce their numbers. A more in-depth discussion of how to move the pendulum and reduce the number of doubters will be covered later in this book.

Organizational readiness can be a bigger hurdle than simply the lack of understanding and poor communications; it can truly be a factor of organizational maturity. A critical error that many make when trying to implement new processes is overestimating the maturity and the readiness of the organization to accept change; in other words, organizations can only assimilate concepts equal to their process maturity. An example of this would be trying to implement a release schedule process into an organization where the norm is to implement systems at will without having to schedule implementations. This organization doesn't have any release process and has ungoverned project management guidelines. If all of a sudden it was mandated that a release schedule be used, the organizational push back would be significant since there is no existing process in place and very little governance. Even though the resources agree with the theory of release scheduling, in practicality, the organization is not ready or mature enough to execute the idea. Now if the concepts of release management are properly phased into this organization and the benefits of each phase are realized, the concept of a release schedule will be readily accepted when introduced. In fact, if the phasing is in sync with the organization's maturing, the organization will be asking for the implementation of release scheduling rather than resisting it.

One of the biggest errors that can be made is to try to push processes on an organization that is not ready for them. The net result will be a lack of acceptance of

© 2010 by Taylor and Francis Group, LLC

the new processes and the setting up of barriers to future efforts. In an effort to try to avoid this critical error, a readiness assessment should be completed to help set a proper baseline. Carefully planning the pace of the implementation, and actively monitoring the adoption and acceptance of the new process using a plan, do, check, act approach will help to assess whether the pace of implementation is correct.

Practical Application

An issue for many who attempt to implement release management is translating the theoretical concepts into practical application. There is a common misconception that every aspect of an industry standard, end-to-end release process must be implemented to be successful and to realize the benefits. The end-to-end process implemented must be in strict accordance with the guidelines set fourth within the recognized industry standards. Any variance from these standards means that the process will not be "compliant." This concept of being compliant and how an organization fits into the cookie-cutter standard can cause many sleepless nights and a lot of gray hair.

In reality, being compliant with industry standards can be a long-term goal, however, it is a goal that should be secondary. The primary goal should be how release management fits into the existing organization and how the greatest benefit can be achieved in the shortest period of time. Realizing the greatest benefits in the shortest amount of time means applying release management concepts in the most practical manner that fits the maturity of the organization. This means understanding the good practices that the organization already has and how can they be built on, and taking industry best practices and incorporating them into the organization practically at the right pace. In the following chapters of this book, how to take industry best practices and apply them practically so as to get the biggest benefit of the organization will be discussed.

Baseline and Metrics

When organizations start down the road of process improvement or implementing new processes, the common practice is to do a quick assessment of what's broken. This quick assessment is usually caused by an isolated incident that caused the organization pain. The normal reaction is "let's fix it and I know how to do it," and off they go to come up with a solution to the problem without doing the analysis to understand the actual root cause. Sometimes this approach works and sometimes it doesn't. If it does work, then the question may come, "How do we know we fixed it?" Many times the answer is, "We know it is fixed because it isn't happening anymore." If the fix isn't successful, a different approach is tried until the right solution is discovered. If the organization

© 2010 by Taylor and Francis Group, LLC

would have taken the time to do a baseline assessment or measurement, the question, "How do we know it's fixed?" could be answered with documented evidence. In the second scenario, the root cause could have been discovered through creating a baseline of the error, which would have allowed a quicker fix to be discovered.

Creating a baseline and using metrics is the only way to demonstrate how effective a process improvement or new process is. As mentioned earlier, to effectively sell the implementation of release management or any other process, an understanding of the benefits derived from the implementation must be understood. Whether these benefits are cycle time reduction, improved availability, or financial, they cannot be articulated unless a baseline is set and measures of improvement are defined. Creating a baseline and release management metrics will be discussed in a later chapter.

Concept Review

This chapter provides a baseline and an understanding of the concepts of release management that will be expanded on in this book. It is important for the reader to understand that implementing release management has many dimensions that need to be considered. Introducing the different facets that need to be considered when implementing release management sets the foundation for further discussion. The high-level concepts discussed in this chapter were:

- Defining release management
- ITIL service management lifecycle
- How release management fits within the lifecycle
- Why there is confusion between release management and project management
- The financial benefits of release management
- Organizational readiness
- Practical application
- Creating a baseline and how metrics should be used to demonstrate successes of release management

The following chapters will build on each of these concepts with the objective of gaining a full understanding of release management and taking the theoretical concepts of ITIL and applying them in a practical manner.

© 2010 by Taylor and Francis Group, LLC

Chapter 2

Basic Concepts

Understanding the basic concepts of release management will provide the foundation on which the practical utilization of release management will be built. Understanding how release management interfaces with various aspects of service delivery, service operation, and all of the Information Technology Infrastructure Library (ITIL) functions within the lifecycle to build a release practice will strengthen the implementation. It is necessary to have a sound understanding of the basic concepts of release management to understand the value they bring to the process. To be effective in practically implementing release management, it is not only important to understand the basic concepts, it is also important to understand the activities that enable them. When implementing an effective release practice there is a need to understand how release activities relate to other developmental processes such as project management, quality assurance testing, technical architectures, and other supporting technical verticals, and having this understanding strengthens and sustains the practice.

Objective and Mission

The basic definition of release management is ensuring that all aspects that are related within an IT service are considered when creating, building, and implementing a new or subsequent release of that service. This definition can also serve as the objective of release management. If this was to be expanded, words such as repeatable, controllable, and scalable could also be used. So if asked what the objective of release management is, the answer would be:

> Release management ensures all related components of an IT service are considered when the service is created or modified and provides activities that are repeatable, controllable, scalable, and sustainable.

© 2010 by Taylor and Francis Group, LLC

This objective will differ from other objectives found for release management in that it gives consideration to the concepts that focus on being able to sustain a quality process, namely reusability, control, and being able to scale release activities. After all, creating a process that cannot be sustained either because it is too complex, too laborious, not right sized, or does not have perceived value, is a waste of valuable resources. In addition to reusability, controllability, and scalability being important components of sustainability of the process, they are also important to controlling the cost of release development.

Within the objective, the phrases "*ensures that all related components*" and "*are considered*" are key in understanding the role that release management plays within the delivery of quality releases and have an impact on the operational stability of the service. Ensuring that all related IT components of the IT service that are being designed or modified are *considered* defines release management's holistic approach to the delivery of services, which enables the business to meet its strategic goals. In a holistic approach, release management functions in several capacities—as an enabler, in a consultative capacity, and through governance; balancing these roles can be different in each organization depending on the culture and maturity. The actual role that release management plays in each organization is one of the first points where there is a departure from the theoretical to the practical.

Creating an organization-based mission statement describes how release management fits within the specific organization. It plays an important role in the assimilation of release management and promotes ownership of the practice. The generic mission statement of release management is:

> Delivering quality, operationally ready releases that will improve the day-to-day operations of IT services enabling the business to meet and exceed their strategic goals and objectives.

This mission statement is simple but calls out the need for the delivery of quality and tested releases. It also sets the expectation that a release of existing IT services will be an improvement to existing functionality and must have tangible benefit to the business. In the most simplistic terms, the mission statement sets forth three significant questions that should be asked when considering whether a release should be done.

Is there benefit to the business? Will the release enable the business to accomplish a strategic goal, increase revenue, improve customer satisfaction, or solve a specific business problem?

Can the release be planned, built, tested, and delivered within the required cost, time, and quality requirements?

Will the new functionality of the IT service improve the day-to-day operation of the business service or will it increase the complexity?

© 2010 by Taylor and Francis Group, LLC

If the answer is "no" to any of these three questions, then the organization should take a serious look at why the release is being built. It is not uncommon for an IT organization to continue down the path of implementing new functionality for a service because there is a perception that it is good for the organization without really understanding how the service is being used. Simply asking these three simple questions before embarking on the creation of a new release can save a lot of time and money. If the answer to these questions is "yes," then a more in-depth analysis should be completed to ensure there is enough return on investment (ROI) to proceed with the release.

A basic concept in creating a strategy for implementing a release practice is creating a vision of what the end practice will look like once it has been created and how this vision is going to be achieved. ITIL introduces a simple strategy model that uses four questions that should be asked when embarking on creating or improving a process: Where do we want to be? Where are we now? How are we going to get there? How do we know we got there?

The first step of this model (see Figure 2.1) asks "Where do we want to be?" These six simple words should cause the organization to really examine their internal processes and create a vision that will drive the creation of the process. These decisions should be agreed upon by key stakeholders since this will drive strategic decisions and directions. Once this strategic decision has been agreed upon, the mission statement and objective can be created and used to communicate the strategy. The mission statement and objective statement should also be used as the guiding principal when creating the process

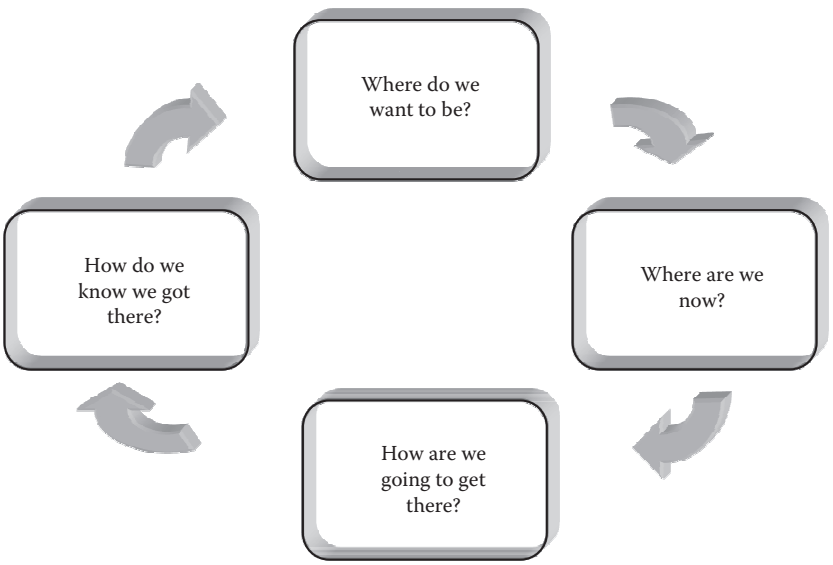


Figure 2.1 Simple strategy model.

© 2010 by Taylor and Francis Group, LLC

and practice; it should be referred to often to ensure that the developmental activities remain aligned with the organization's strategy.

Release Policy and Procedures

Organizations embarking on the creation of a release management practice will need to create a documented policy and procedure that resources using the process will be able to reference to understand what is expected and what processes need to be followed. The release policy defines the scope, strategy, and standards of the release practice within the organization; it is the playbook for delivering a quality, operation-ready release. The release policy should be considered a living document and will continue to be revised and improved as the release practice continues to mature and as the organization assimilates the release process. Care should be taken when creating a release policy not to include concepts and standards that cannot be implemented due to lack of organizational maturity or readiness. In the same vein, however, those concepts and standards that need to be implemented must be included within the policy. An example of a release policy can be found in Appendix A.

There is a saying that you don't want to throw the baby out with the bathwater. Organizations that have successfully or have at some level been successful in delivering releases will have some good release practices that should be retained and incorporated into the release processes being developed. The practices that have been used by the organization may not be consistent or documented. These practices should be reviewed to determine their viability and what value they provide. Once this initial assessment has been done, these existing processes can be the starting point for process development and the basis for the standards that will be documented within the release policy. This is the second part of the strategy model—*Where are we now?* In addition to creating a basis for the creation of standards, using existing processes in some form will lead to fewer changes and create better acceptance of the new processes being introduced.

When creating a release policy or other deliverables within the process, three questions should be continually asked:

- What value does this provide to the user?
- Who is going to use the policy?
- How is it going to be used?

We have all seen processes that appear to be meaningless and the value they bring is questionable. These processes usually break down and fail. Being able to clearly articulate the value proposition of the process will increase the adoption.

Following this approach, the first thing the release policy must address is the purpose and the value of release management. Being able to document the defined strategy and objective, the guiding mission statement, and the defined scope will

© 2010 by Taylor and Francis Group, LLC

set the basis for the policy. The release policy should also include the organizational context for release management; this context will define the role release management plays within the organization and the developmental process. There are three roles that release management plays within the organizational context—guidance, facilitator, and governance.

Guidance

One of the primary functions of release management is to define the release process, manage the release, and provide guidance throughout the development of the planned release cycle. In a later chapter the use of the release lifecycle (RLC) will be introduced and discussed. The release lifecycle is a systematic approach that defines and provides a roadmap of the checkpoints and deliverables that need to be produced to provide a value-added release. Consulting with design teams throughout the delivery process provides for successful releases of applications and associated hardware. Release management works closely with the project management office (PMO) to provide training for the project manager's processes and the key deliverables throughout the RLC.

Facilitator

Once the release process has been established and implemented, there will be multiple checkpoints and deliverable reviews that will take place. These reviews will be managed and conducted by the project manager with release management to ensure the required activities and tasks are being completed to ensure the release schedule is being maintained. These reviews should be focused only on release deliverables and tasks; deliverables required by the PMO should be reviewed by the PMO and excluded from this review. Release management helps to facilitate these reviews and to identify any issues or conflicts between competing releases. Release management should have an enterprise view of all releases and can facilitate a review with competing project teams to assist with the resolution of any issues arising from scheduling conflicts.

Governance

The governance role within an organization is either a role that everyone wants or no one wants. The role release management plays should be clearly defined within the organization and documented in the release policy to ensure a full understanding. Generally, governance within the release realm pertains to the tasks related to the quality delivery of the release and the ability of the organization to support and operate the service to the designed service levels. These tasks and deliverables include, but are not limited to, different levels of testing,

© 2010 by Taylor and Francis Group, LLC

support documentation, service level agreements, and service continuity plans. These deliverables and tasks should be right sized for the specific release and described in the RLC.

Another governance role release management can play is in the area of regulatory compliance. In every industry and in every country there are specific regulations that need to be met, whether it is Sarbanes–Oxley Act (SOX), Health Insurance Portability and Accountability Act (HIPAA), Federal Deposit Insurance Corporation (FDIC), or JSOX, the Japanese version of SOX, just to name a few, and the release process can be designed to assist in complying with these regulations. How this can be done will be covered further in the release lifecycle.

Release Standards

The release policy and procedure document should also provide direction on standards and guidelines that need to be followed. The guidelines described in the release policy need to be created to ensure that release development aligns with the goals and objectives of the organization. The document should be generic enough so that it doesn't have to be revised frequently, but specific enough to provide the user enough information to use the process. These standards, guidelines, and policies must be created to fit the organization where they are being used. Many generic standards and definitions can be found in various publications; however, to be successful they must be tailored to the specific organization. The concept of the release lifecycle, which was introduced earlier in this chapter, will help development teams understand and use the standards, policies, and guidelines documented in the release policy.

Basic Concepts

The basic concepts of release management should be incorporated into the release policy. However, before they can be included in the policy, the concepts need to be customized to fit the specific organization. A basic understanding of these generic concepts is needed:

- Release models
- Release schedules
- Naming conventions

Release Models

Being able to classify and categorize different types of releases into release models allows one to determine the types of governance and review that should be completed. It also provides for the right sizing of release structure, deliverables,

© 2010 by Taylor and Francis Group, LLC

and promotion and implementation methods for the release. Release models also provide the organization with a common language and provide a connection to the project management methodology.

While there are many considerations in developing release models, one of the basic considerations is what Configuration Items (CIs) or parts of the IT service should be released together. Being able to determine the dependencies between CIs within the IT service will determine what parts of the IT service need to be released together and understanding these relationships define the release unit. As illustrated in Figure 2.2, release units can be as small as a single module or as large as the entire IT service. These different types of release units can be separated into three categories: delta, full, and package.

- **Delta release**—A basic release unit that usually includes a small number of modules or components of the IT service that have changed since the last full or delta release.
- **Full release**—A release that is comprised of several components of an IT service and may include several delta releases. All of the components are built, tested, and implemented together.
- **Package release**—When several releases are grouped and implemented together, a package release can be comprised of a full release and delta releases.

Another consideration in creating release models is release type. Release types describe the complexity of the actual change or release being implemented. The generic description of release types are: minor, major, and emergency. Organizations should look at their existing methodology to determine if release categorizations

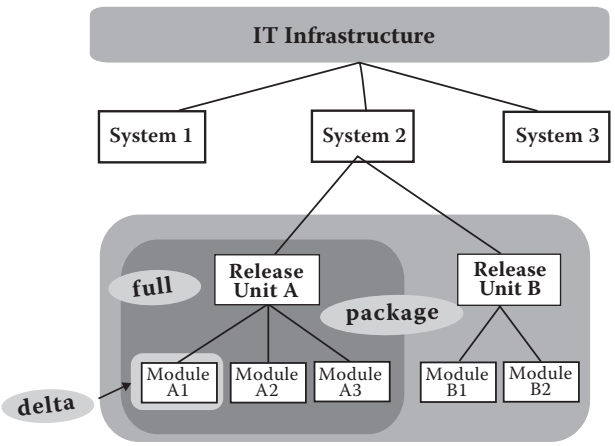


Figure 2.2 Release units.

© 2010 by Taylor and Francis Group, LLC

already exist. If so, then consideration should be given to using the existing naming convention. Using existing naming conventions will improve the acceptance of the release methodology. Generic release types are relatively easy to define.

- **Minor release**—Consists of small enhancements and fixes; some minor releases may have been completed as emergency releases. Minor releases will supersede all previous emergency releases. More frequently done than major releases, minor releases can range from once a month to quarterly.
- **Major release**—Contains significant or large portions of new functionality, major upgrades, or new service implementations. Major releases will supersede all minor and emergency releases. Less frequent and requires significant planning and lead times. These are usually done only once a year.
- **Emergency release**—This type of release is done in response to an incident or significant problem and may be related to the emergency change process. Typically the release is small and similar in nature to a minor release, but is completed in much less time.

Examples of organization-specific release types could include categorizations such as projects and system enhancements rather than major and minor releases. If an organization has classified their release types this way, then this naming convention can be used.

Deployment Considerations

An additional consideration for a release model is the deployment options for the release. The type of deployment model used is dependent on many factors, such as complexity of the release, the impact the implementation will have on business operations, and resource allocation needed for implementation. When deciding which type of deployment model to use, a risk assessment should be done to determine which model presents the least acceptable risk. The various deployment models range from a big bang approach to a phased approach.

- **Big bang deployment**—The big bang model is the riskiest deployment strategy. It is simply implementing the release all at one time for all locations and all users. This presents the greatest risk as it is all or nothing. If issues occur with the release deployment it affects all users in all locations.
- **Phased deployment**—The phased model still has risk associated with the deployment, however, not as great as the big bang model. The phased deployment model involves creating specific groups of users and deploying the release to each of the groups in a sequential order. The deployment groups can be grouped by types of users, location, or function. A phased deployment can be done over an extended period of time or over a shot period.

© 2010 by Taylor and Francis Group, LLC

- **Phased parallel deployment**—This approach is a variation on the phased deployment mode. The difference is the running of parallel systems during the deployment period. In other words, the legacy system continues to function and the new system is deployed in parallel. This is the most complex approach since there are two systems running, performing the same service. Both systems have to be supported and maintained.
- **Pilot deployment**—The pilot deployment model presents the least risk of the deployment models. In this approach a pilot group is identified and the release is deployed to this group for a specified period of time. During this period of time, the release is monitored closely to ensure the release is successful and addresses any issues that arise. A modified approach is to start with a small alpha group, expand to a larger beta group, and then deploy the release to the targeted group.

The deployment model utilized needs to be considered carefully to ensure the appropriate risk matches the complexity of the release and the risk the organization is willing to accept.

Release Schedules

A key benefit of release management is being able to determine and schedule when a new IT service or enhancement will be implemented into the production environment. Being able to plan and schedule new functionality allows for better resource utilization, increased quality through planned testing, reduced financial cost, and greater implementation success. Another by-product of good release planning is the increase of customer satisfaction with the customer and end user.

There are two types of release schedules. The first is the enterprise release schedule, and the second is the release schedule of a specific service; both play a part in creating a successful release practice.

Enterprise release schedules are used to gain a holistic view of when all releases are scheduled within the enterprise. This type of release schedule is used primarily for the scheduling of development resources and financial planning. An enterprise release schedule should include, at minimum, a view of releases that are scheduled for the next twelve months, ideally, eighteen months to twenty-four months. Not only should an enterprise release schedule include releases by service, it should also include releases of cross-functional IT services used to support the business services and applications. An example of a release schedule would be: version 1.0 of a database is scheduled for May and an upgrade to that database to version 2.0 is scheduled for June of the following year. While this is important for the reasons previously stated, it can be more important for application teams to understand which versions of infrastructure or technology are being used since applications are typically built to use specific versions of technology products. If there is a change in version, the functionality of that technology product may change and the existing

© 2010 by Taylor and Francis Group, LLC

application functionality may not be able to work properly or may be disabled. Therefore, it is extremely important to understand when the underlying infrastructure of the enterprise is changing.

As illustrated in Figure 2.3, an enterprise release schedule can be as simple as a spreadsheet with products and dates or as complex as detailed release plans.

Release schedules can play an important role in financial and strategic planning in terms of providing a roadmap showing when expenditures will be needed and providing the timing of capitalization of the assets being developed.

The second type of release schedule is related to the individual release of a specific service or application. While the common practice is to relate a release to a specific application or CI, consideration must be given to how that release will affect the specific service and related services. This type of release schedule can be called a *service release schedule* or a *product release schedule*. Typically, these types of release schedules contain both major and minor releases of the specific product or service. Major releases are typically scheduled once every twelve to eighteen months and involve significant planning and development. Minor releases can vary when they are scheduled from every thirty days to every six months; much of this is dependent on the stability of the service or product. A newly implemented service will typically require more frequent minor releases and as it becomes more stable, will require fewer minor releases.

When planning a service release schedule, several factors must be considered, including timing, resources, cost, funding, and business need. In some companies, system enhancements are implemented at the whim of the associated business unit or customer and are not scheduled. When this happens, the time to delivery may be quicker, however, typically the delivery cost is increased significantly and the quality is reduced due to inadequate planning and testing.

The release policy should include a couple of release schedule models to help the service owners understand the different time frames that could be used in developing their release schedules. Figure 2.4 provides some sample release schedule models.

An enterprise release schedule will be created when the individual service's schedules are rolled up into a holistic schedule for the enterprise. Both will be managed by release management; each at different levels. This is a very simplistic explanation of what a release schedule is, however, it gives the basic premise of understanding and helps to point out that there is no need to overcomplicate the creation.

Naming Conventions

The concept of using a consistent naming convention of release versioning is a simple concept that is sometimes used and sometimes is not, or if used, may be used inconsistently across the enterprise. This is not the use of catchy names to identify the newest application; rather it is the use of a consistent naming approach for version control. A consistent naming convention allows all resources to understand the

© 2010 by Taylor and Francis Group, LLC

Release	*Jan.*	*Feb.*	*Mar.*	*Apr.*	*May*	*June*	*July*	*Aug.*	*Sept.*	*Oct.*	*Nov.*	*Dec.*
e-Mail		MGR		MR		MR		MR		MR		MR
CRM	MR		MR		MR		MR		MR		MGR	
Receivables System	MGR	MR	MR	MR		MR		MR		MR		MR
Database Upgrade	MR		MR		MR		MGR		MR		MR	

MGR - Major Release

MR - Minor Release

Figure 2.3 Enterprise release schedule.

© 2010 by Taylor and Francis Group, LLC

Release Model	*Program*
Mature	■ Quarterly minor releases ■ Major release 10 months
Standard	■ Minor release 6 weeks ■ Major release 12 months
Unstable or new	■ Minor release 4 weeks ■ Major release 8 months

Figure 2.4 Release schedule models.

status of the product; earlier versions are typically simpler and have fewer features and less functionality than newer versions. Additionally, being able to identify the version of a product allows for an understanding of the features and functionality so other products or services know what that version contains from a developmental perspective.

An organization needs to make a determination of how the nomenclature of the naming convention of versioning will be used. Figure 2.5 is an example of a naming convention schema.

Naming Convention	*Designation*
1.x	Major Release
x.1	Minor Release
x.x.1	Emergency Release

Figure 2.5 Naming convention schema. (ERP system x,x).

Using the example in Figure 2.6, the initial release of the new enterprise resource planning (ERP) system would be version 1.0. Of course this was a major release since it was the initial release of the product. Since this is a new system, there will be a need to do minor releases more frequently. In this example, minor releases were scheduled every four weeks so bugs can be repaired. The first minor release is scheduled and named, ERP 1.1, within four weeks of the major release. The second minor release was scheduled eight weeks after the initial implementation and was named ERP 1.2. However, in the sixth week, a major bug was discovered that caused a major incident, and an emergency release was created and implemented; this release was named ERP 1.1.1. Even though an emergency release was implemented in week six, work still continued on the minor release that was scheduled for week eight. When week eight arrived, minor release ERP 1.2 was implemented. Finally, twelve months from the original implementation, it was decided that

© 2010 by Taylor and Francis Group, LLC

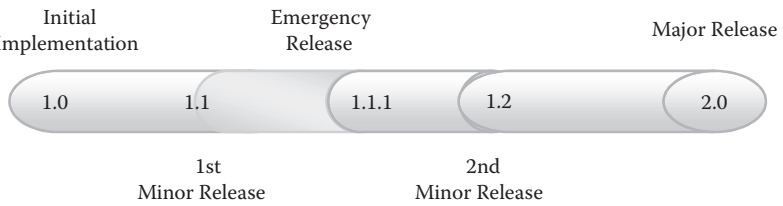


Figure 2.6 Naming convention example.

significant functionality was going to be added to the original base product, and a major release was scheduled and named 2.0.

Release Procedures

The release policy needs to include operational guidance for the organization's resources that are using it. The guidance that should be included in the release policy should be focused on providing a high-level roadmap on how to use and navigate the release process. It should include information about the roles and responsibilities of the "actors" involved in the process, such as the release manager, release project manager, the quality assurance resources, and others involved in the release of a service. The release policy should also provide information about the release lifecycle, the roadmap for delivering quality releases. Finally, the release policy should provide instructions on how to or what criteria needs to be followed to execute an emergency release. While the entire release policy is focused on how to deliver a release, the procedures for an emergency release should be clearly understood since the planning, governance, and execution may need to be completed in an expedited time frame and my happen outside of normal business hours.

Emergency Releases

There will be situations when it will be necessary to create, build, test, and execute a release outside of the normal release process. Usually this will be in response to an incident where there is a loss of service. It is advantageous to establish a process to maintain the integrity of the IT environment and to ensure that any release implemented maintains the quality standard that has been established through the normal release process. This quality standard can be maintained by creating a process that ensures accurate impact assessment and correct testing prior to implementation. A key part of creating an emergency release process is to understand what level of risk the organization is willing to assume by not following the normal process. This same level of risk acceptance will align with the risk that is acceptable during the emergency change process.

© 2010 by Taylor and Francis Group, LLC

In normal conditions, a release must be approved though the Release Control Board (RCB); however, during an emergency release, the approval of the release may be delegated to the release manager or other managerial oversight. There should be some type of post-implementation review done and reviewed with the RCB to ensure awareness and after-the-fact approval. Additionally, an acceptable level of testing must be completed prior to implementation. This level of testing will not be as extensive as the normal level of testing would be, but it should be focused to ensure that the emergency release will be successful and not cause additional loss of service.

Although an emergency release will be done in an expedited manner, it is still necessary to understand the impact the release will have on other components within the service being changed. This can be done by reviewing the configuration management database (CMDB) or other sources that have service relationship information. The emergency release must also have a documented implementation plan that includes implementation requirements, tasks, and a back-out plan. These activities should also align with the emergency change process that will also include the preparation of an emergency request for change (RFC).

Roles and Responsibilities

As with any process, to be successful, the roles and responsibilities of the resources involved in planning and executing the process need to be clearly defined and documented, and the release management process is no different. An effective tool used to document the roles and responsibilities is a RACI matrix—responsible, accountable, consulted, and informed. A RACI matrix lists the tasks or activities of the process and the role that will have some type of involvement in performing that task.

- R—*Responsible* for performing or completing the task or activity. There may be more than one role that is responsible for completing a task or activity; therefore, there may be more than one "R" listed for the task.
- A—*Accountable* for the proper execution of the task or activity. The role designated as an "A" has sole accountability for the activity. There is only one "A" per task.
- C—*Consulted* during the execution of the task or activity, this role may provide information needed to complete the task. There may be more than one "C" per task.
- I—*Informed* of the task or activity; purely informational. This role has a need to know the task or activity is occurring, but does not need to be directly involved in the execution of the task. There may be more than one "I" per task in the matrix.

© 2010 by Taylor and Francis Group, LLC

When completing a RACI matrix, not all of the roles need to have a designation in the matrix; there will be some tasks in which some roles will not have any involvement. However, there must always be only one "A" per task or activity.

The roles commonly identified in the release process are the release manager, release process manager, release project manager, release coordinator, quality assurance, and release control board. Depending on the maturity of the organization, other roles that may be included are the service owner, change manager or management, configuration manager or management, and other service delivery roles. If there are roles already established within the organization that do not fit the suggested roles, then it is acceptable to add them to the matrix. It is also acceptable to use a different role name if the activities being performed by a role have a different name. The organization should make the process their own—if using different names will do that, then do it.

- **Service manager**—Has overall responsibility for the service delivery process and functions. Engages senior management and is the organization's chief process advocate.
- **Release manager**—Ensures the release process is executed as designed. Is a "go to" role through which resources performing the release process can gain guidance about the process. Schedules and leads the release control board meeting and other release review meetings. Creates and manages the release schedule.
- **Release process manager**—Creates, audits, and improves the release process. Ensures the release process aligns with other development and service delivery processes.
- **Release project manager**—This can be a dual role played by the project manager that may report to the project management office. Has the responsibility for the delivery of the individual release and ensuring that required tasks and deliverables are executed according to the release project plan.
- **Release coordinator(s)**—Assist the release manager in the day-to-day operation of the release process. Can be assigned to a specific release to assist the release project manager with the release deliverables.
- **Quality Assurance (QA)**—The QA role is responsible for ensuring that the testing strategy is appropriate for the level of release. Ensures that the testing strategy maps back to the business requirements and that the agreed-upon testing requirements have been met.
- **Release Control Board (RCB)**—Primary role of the RCB is the oversight of the enterprise release schedule, which includes the approval of releases and the rescheduling and deferring of releases. The RCB can also act as a governing board if there are conflicts in release scheduling. The RCB can work closely with the Change Advisory Board (CAB) or can be the same body.

© 2010 by Taylor and Francis Group, LLC

The roles of other processes can also be included and the relationships among other service delivery processes will be discussed later in this chapter. A sample RACI chart showing common tasks and roles is located in Appendix B.

Release Lifecycle

At the heart of the release process is the release lifecycle (RLC), which provides the roadmap for delivering successful quality releases into implementation. It begins at the point where a release is approved and provides guidance through several phases that end with a post-implementation review of the implemented release. The release policy should include an overview of the RLC and the purpose of each phase. The different phases of the RLC are:

- Initiation
- New Environment Request (NER)
- Release configuration and build
- Quality Assurance (QA)
- Operations Readiness (OR)
- Implementation
- Post-implementation

While different variations of the RLC can be found in other publications, these other publications do not provide an in-depth understanding of how to execute on the RLC. Chapter 5 is dedicated to providing an in-depth view of the RLC and how to use it.

Definitive Media Library

In a good release management practice, original code and copies of software are stored in the definitive media library (DML). The DML can be a combination of logical secure storage, a locked file cabinet, or a combination of both. In today's environment there are tools available that can be used to store electronic copies of software and data, and copies can be "checked out" or borrowed for use. The idea behind a DML is that a copy of a software can be checked out and worked on; however, the "gold" copy original is still intact in the DML in case anything were to happen to the checked out copy. Once a newer version of the software is released into production, a gold version of that release is also stored in the DML. The release manager is responsible for the DML and development teams should go to the release manager to check out a copy. If a DML exists within the organization, procedures on how to access the DML can be included in the release policy.

© 2010 by Taylor and Francis Group, LLC

Measures and Metrics

The release policy should also include information that will be used to measure the success of the process and the success of individual services. The metrics that are used to measure success can vary and should be structured to the needs of the organization. Some organizations may want to measure the time it takes to complete a release, while others may want to measure the quality of the releases being implemented. In truth, both of these measures are valuable measurements and some version of these metrics should be used.

Another factor in creating metrics is the need to set a baseline from which a comparison can be drawn; without a baseline, it will be impossible to determine how successful the process has been. Chapter 6 will go into more detail about what type of measurements and metrics can be used in the release process and how to use the metrics once they are captured.

Release Management Activities

Earlier in this chapter the basic concepts of the release lifecycle were introduced; the bases for the release lifecycle are the activities that need to be performed within the release process. These activities, while documented as good practices of ITIL release management, are good practice when applied to any developmental process, whether it is IT related or non-IT related. The activities focus on organization, planning, preparation, quality assurance, and implementation.

- **Release planning**—Part of the strategy model involves the questions where do we want to be and how do we get there; release planning answers these two questions. Being able to create a sequenced plan of what is to be achieved with the release, what tasks need to be completed, what resources are needed, what the budget is, and how the release fits into the release schedule. Creating a comprehensive project plan for the release is the foundation of a successful release.
- **Building the release**—The actual building of the release may span many areas of the IT environment depending on the scope of the release being developed. Using the release project plan created in the planning phase, the release project manager can start to execute against the plan. Engaging resources from infrastructure, application, and other technical areas as needed, the release is constructed in a controlled manner. Initial testing is done during this phase to determine the quality of the basics of the release.
- **Ensuring fit for use and purpose**—Fit for use and fit for purpose are key measures of whether the release meets the objectives established during the planning phase. To determine whether the release meets its objectives, a direct mapping of the business requirements to the new functionality should

© 2010 by Taylor and Francis Group, LLC

be done. A second way to determine if the release meets the expectations is through user acceptance testing.

- **Operational readiness**—An integral part of the release is how it will perform and be supported after it is put into production. Operational readiness is the activity that tests and measures this. The process encompasses infrastructure readiness, knowledge management, support readiness, and user acceptance.
- **Preparing for release implementation**—The activities needed to be completed to prepare for the implementation of the release are more process related than technical. These activities include communication of the release to the business, training the users and support staff, ensuring that release documentation is properly stored, and finalizing the implementation date.
- **Release implementation**—While the responsibility of the implementation of the release falls on the implementation team, the release manager or the process monitors the release to determine the success of the implementation.
- **Post-implementation**—After the release is implemented, a post-implementation review (PIR) needs to be completed to ensure that all objectives of the release plan have been met. The objectives that were not achieved should be the basis for future minor releases to the service. In the spirit of continual service improvement, a lessons-learned session should also be conducted to determine how the process can be improved. The areas identified need to be documented and entered into the knowledge management database for future reference.

Developmental Relationships

The creation and delivery of IT services entails a multitude of different technologies and ways to develop and implement them. Release management ties many of these processes together and brings organization, focus, and quality to the delivery. The release process does this by taking a holistic view of the delivery process. In taking a holistic view, the release process interfaces with the planning, approval, build, test, change, implement, and operating processes. The level of involvement in each of these different phases can differ by organization and process. Generally, the release process has a greater involvement and dependencies in the development and implementation processes and less in the approval and actual operation of the service.

As with any process, there is an entry point and an exit point, and this also applies to the release process. The entry point for the release process can vary for each organization, however it should be at the point when discussions about how the release is going to be developed are occurring. This can be during the approval process or after the release has been approved through another process. However, it is essential that release standards are incorporated into any development plan; these standards will improve the quality of the service once it is delivered. The exit point from the release process is often after the post-implementation review is completed;

© 2010 by Taylor and Francis Group, LLC

however, it can be argued that the release process really never ends, especially if a continual service improvement process (CSIP) is being utilized. The release process needs to be part of the improvement cycle of CSIP.

In understanding the value of having a good release process we need to look at the relationship the process has with other standard developmental processes. These relationships can be the inputs or outputs to the process or the dependencies of either the parent or child process of the release process.

Approval Process—Gaining approval to develop the service or enhance it is where it all starts. There are various processes for gaining approval. Usually they entail the creation of a business case that describes what the service will do, what benefit will be derived, and the ROI of the service. The release process brings value to the approval process by creating a standardized delivery process that provides a roadmap to the task, activities, and the resources that will be needed to deliver the service. This roadmap takes the mystery out of what needs to be done. As the release process matures, it will be able to provide the cost of tasks and activities and the time it takes to complete each.

Design Process—The basis of a successful service implementation is a quality design process that is clear, concise, and fully captures the requirements of the customer. Taking the business requirements, translating them into a technical design, and creating a project plan or release plan is all part of the design process. The release process has more of an influence in the design phase and creating the release plan; it also guides the development of the technical design completion and the creation of the testing plan that will be used to ensure the quality of the release. This is not to say that release management creates the technical design or the test plan; on the contrary—this is the role of the technical teams and the quality assurance group. Release management provides the process, templates, and oversight to ensure the right resources have provided the necessary input to create a quality release. The groups that utilize the process will be the groups that are actually going to perform the activities and tasks. These groups may include server teams, networking teams, database teams, the organization's security group, and other service management processes. Creating a collaborative design document will ensure all involved processes will have the full holistic view of the service or product being created.

Creating the governance structure during the design phase will set the expectation that the prescribed task and activities need to be completed properly before moving to the next phase. Creating a test plan mapping directly to the requirements will ensure a quality release and identify issues early in the process resulting in less rework later in the process; rework is costly and very resource intensive. Another role that release management plays is ensuring that an acceptance criterion is established in a collaborative manner with the customer and IT. This acceptance criterion must be met and approved by the customer prior to the release going into production.

Creating a release plan that calls out these activities sets the timeline and date of delivery for the release. Once this is done, a realistic expectation can be set with

© 2010 by Taylor and Francis Group, LLC

the customer. The release plan will also identify the dependencies and relationships between the developmental groups needed to deliver a quality release on time and within budget.

Build Process—As the release progresses through the different phases of the build process, the release project manager will be monitoring the agreed-upon deliverables and checkpoints of the release. The main focus of the release during this phase is to ensure the agreed-upon activities are being completed as prescribed and the quality of the deliverables is acceptable. The release project manager will work with the project team to set up checkpoint meetings to ensure the milestones of the release plan are being met. As with the design process, the release process does not replace the teams building the release; rather, it allows the development teams to do what they are good at and provides a mechanism for cross-team alignment.

It should be noted that the release project manager may be a dual role played by the project manager that may have an alignment to the project management office or they may be different resources. At this point it may sound as if the release process is replacing the organization's project methodology; on the contrary, the release process is aligned with the project methodology and supplements it. Chapter 3 is dedicated to understanding the differences and similarities of project management and release management.

Test Process—A primary goal of a good release process is the delivery of a quality release that improves the customer's ability to increase their productivity and revenue generating abilities. The way this is done is by ensuring the appropriate testing is completed, reviewed, and approved. Once again, the release process does not create the test plan, decide on what type of testing should be completed, or complete the testing; the release process is responsible for ensuring the testing strategy agreed upon during the design phase has been executed and that results are acceptable.

It is not unusual during testing that defects are discovered. The magnitude of risk these defects pose to the release must be classified, and the high-risk defects must be mitigated prior to the release being implemented. Moderate and low-risk defects need to be reviewed to determine if the release can go into implementation and be corrected in future releases or if they should be corrected prior to implementation. It is the role of the release process to ensure these defects have been identified, logged, and reviewed by the customer.

There are different methodologies and types of testing that should be completed. Of course, release management is interested in ensuring that the testing of the release functionality is completed and performs as designed. In addition, functional testing, performance testing, security testing, and user acceptance testing are critical to the operational aspect of the release and must be completed, documented, and approved by the customer prior to implementation. These types of testing should not be deferred or avoided; not doing them could have a significant impact on the performance and operation of the release. Insufficient security testing could allow vulnerabilities to the environment that could be exploited and

© 2010 by Taylor and Francis Group, LLC

cause significant harm to the organization. The execution and review of various testing standards within the release process will significantly increase the reliability, usability, and quality of the release being implemented.

Implementation—Once again, the release process focuses on ensuring that critical tasks and activities are completed rather than actually performing the task. Cross-functional alignment, coordination, and relationships are critical during the implementation phase to ensure smooth transition and to reduce any incidents or loss of service during the execution. Tasks on the release plan will ensure that an implementation plan with detailed task and a timeline is completed and that a back-out plan is also created if the implementation fails. An important aspect of the implementation plan is graduated success criteria that must be met prior to moving to the next phase of the implementation. Release management should review the implementation plan to ensure the success criteria are defined.

Another aspect of implementation release management will be involved in is ensuring the proper training of resources has been completed, including the service desk. Ensure all release notes and documentation are stored in the appropriate repository and the code being implemented is logged into the DML. An important aspect of implementation is to ensure the release has received final approval from change management prior to being executed.

Operational Considerations/Total Cost of Ownership

An important aspect of the release process that is often overlooked is how following a good release process affects the operational aspect of the service and reduces the overall cost of ownership or operation of the service. While it is easy to see how a good release process can improve the quality and reduce the cost of the delivery process, the benefits do not stop once the release is implemented. The simple and quick view is that the quality of the release will be improved due to better governance, development cost will be reduced due to efficiencies gained through a consistent process and better alignment of resources, and reliability will be improved through better testing. However, reduction of operating cost benefits due to a good release process does not stop at implementation. Release management takes a holistic approach to the delivery of the service from the original creation, to the operation, to the improvement of the service through subsequent releases, and ending with the retirement of the service. By taking a holistic view, aspects such as reusability and continual service improvement help to reduce the cost of ownership of the service.

During the release lifecycle, deliverables are created and executed that improve the operation and support of the release while in production. Many times project management is more concerned with the delivery of the project rather than the operation of the service; therefore, operational considerations are overlooked or left to the final moment and not adequately addressed. Operational considerations and

© 2010 by Taylor and Francis Group, LLC

deliverables, such as early life support of the service during implementation, support and escalation documentation used by the service desk, knowledge articles for the knowledge base, FAQs, system diagrams, and service restoration plans, must be delivered to provide the level of operation the customer deserves.

Concept Review

This chapter has introduced many concepts of the release management process; these concepts provide a baseline understanding and set the foundation of a common language that will be used during the creation and execution of the release process. In this chapter the basic concepts were reviewed and the values of these concepts were discussed. The concepts reviewed in this section were:

- The objective and mission of release management was discussed.
- Use of a strategy model to ensure alignment with the organization's strategic goals and alignment with IT strategy was examined. The strategy model will also validate whether the planned release will show value.
- Creation of a release policy and procedures document to provide a reference for the organization was described. The release policy provides standards, expectations, and roles and responsibilities of the process.
- The release policy should be tailored to fit the organization's maturity and acceptance level. These levels should be challenged and pushed to improve.
- Organizational context of release management can be aligned to three categories: guidance, facilitator, and governance.
- Release models need to be created to help the organization determine what types of governance and review should be completed. They provide for the right sizing of release activities and implementation.
- The three release types were defined—delta, full, and package.
- Release categories were defined—minor release, major release, and emergency release.
- The need for and the creation of an emergency release process and when it should be used was examined.
- Enterprise release schedule and individual release schedules need to be used to aid in the planning of releases and improve the financial and resource planning.
- Utilizing a common naming convention for versioning of releases was examined.
- Roles and responsibilities of resources within the process were defined; the utilization of a RACI matrix should be used to provide a clear understanding.

© 2010 by Taylor and Francis Group, LLC

- The release lifecycle was introduced; its utilization improves the time to delivery and quality of releases. The RLC defines and provides guidance on the activities needed to be performed within the process.
- Release management interacts with the entire developmental cycle and has touch points to many existing processes.
- To determine the success of the release process, it must be measured and metrics that demonstrate value should be created.
- While the generic concepts provide a good basis, these concepts must be adapted to fit the maturity and culture of the specific organization to create a successful process.

© 2010 by Taylor and Francis Group, LLC

Chapter 3

Release Management and Project Management: Kissing Cousins

The organization has now matured to the point where release management can be introduced, a proposal is prepared, and it is presented to the Chief Information Officer (CIO) and IT management; the feedback is, "We already have project management. It looks like release management is the same thing. Why should we duplicate our efforts?"

This reaction is not unusual. There is a general confusion that release management and project management are one in the same, and to the untrained eye there are many similarities. However, there are distinct differences in goals and objectives. While some deliverables can be used for both processes, the way they are used differentiates the two. Conversely, there are deliverables that are unique to each of the processes. Being able to understand and articulate the differences, similarities, and complementary likenesses will increase the ability to sell the benefits and need for release management.

Differences

On the surface, release management and project management appear to be the same process; however, there are differences between the two. Understanding how they differ and what role each plays is important in implementing a successful process as well as selling the benefits of each. It is valuable to understand the

© 2010 by Taylor and Francis Group, LLC

dissimilarity between release and project management. Looking at three specific facets—objectives, focus, and goals—will provide a view of the differences.

Objectives. The differentiator between release management and project management is their primary objectives: release management has the objective of delivery of the entire lifecycle of the service, and the objective of project management is the delivery of the agreed-upon requirements of the approved project within the parameters of time, cost, and quality. While both have the objective of quality delivery, many times project management may sacrifice quality to deliver the respective project on time and within budget. Release management, on the other hand, is concerned with the quality and operation of the overall service from inception to retirement of the service.

Focus. Focus is similar to the objective; the difference is that focus helps to bring the objective into application. With this in mind, the differences between release and project management become clearer. Release management is concerned with the holistic delivery of the entire service from the inception of the service to the retirement of the service, and the focus of project management has defined requirements with a start and finish. Taking these concepts deeper provides an understanding of the purpose and reason for different types of deliverables for each process.

One concept that clarifies the difference between release management and project management is that there can be many projects in one release. This is especially true if an organization considers activities of various development groups as separate projects that feed into a parent project. If this is the case, there could be many project managers. If this is the situation, then release management and the release project manager will be a central point for ensuring release deliverables are collected, reviewed, and approved. Projects are integral to the release process; however, many times the project's focus has a start and an end; once completed the project is over. Conversely, the focus of a release is preparing the release for quality operation of the service. These activities include knowledge management, early life support planning, and support documentation. These items are usually secondary to the project completion and sometimes left to the end of the project or delayed until after implementation.

Goals. Objectives, focus, and goals all sound similar, but from what we have seen so far there are differences. Goals are the next step in the cycle of setting the expectation of where we want to be. A common task with any activity or process is goal setting; without defined goals there is no direction. Goals can be specific or general. The more specific the goal, the more focused the results will be. The general goal of release is to ensure that all releases, whether new or supplemental, to a service are operationally ready for implementation and will improve that service. The goal of the project is the quality delivery of the defined requirements on time and within the specified budget.

© 2010 by Taylor and Francis Group, LLC

Similarities and Complementary Likeness

To say there are similarities between release management and project management is an understatement. The similarities between the two processes are the reason there is a common misconception that they are one in the same—just different names from different standards. The confusion is mainly founded in the similarities of the task, activities, and artifacts performed and created. While the activities to create an artifact are used for both processes, which is encouraged for efficiency purposes, the utilization of that artifact can be different for each process; an example is the use of a quality assurance report. Project management would use the QA report to determine if defects exist in the construction of the application or infrastructure component for the purpose of correcting the defect before going into production. While this same information is important for release management, the same information can be used to identify potential points of failure that could occur once the release is implemented into production. The difference is that project management uses the information in a diagnostic manner that will only be used during the project lifecycle, where release management will use the information in a proactive manner and will use it throughout the life of the service until the bugs are corrected or until the service is retired.

These types of similarities exist throughout the development and project lifecycle. It should be remembered that the key difference is how the information and artifacts will be used—project management will only use the information and artifacts for the length of the project and release management will use it for the life of the service. The reusability of the information and artifacts is a benefit of release management. Reusability reduces development time and the cost of future releases simply because development teams do not need to recreate the artifact from scratch to understand the existing version of the release. When a new release is created, the development team simply refers to release notes and documentation from the previous release.

In the same way that there are activities, tasks, and artifacts that are created and used for both project management and release management, there are also activities, tasks, and artifacts created for one process that complement the completion of other artifacts for the other process. An example of a complementary task is the creation of the business requirements document (BRD), which is a project management task. The BRD feeds the creation of service level requirements, the service offering, and finally results in the creation of a service level agreement between the customer and IT operations. The service offering is also a critical component for creating the organization's service catalog. In this example, the BRD is not a requirement of the release lifecycle, however the service level agreement is, and without understanding the requirements and expectations of the customer, the service cannot be built and operated to the customer's needs and expectations.

© 2010 by Taylor and Francis Group, LLC

The release and project management processes do have similar and complementary tasks, activities, and artifacts as explained, however there are some activities of each process that are not valued by the other. It is not that these activities and artifacts are not value added; it is simply that they do not have relevance to the goals and objectives of the related process. It was discussed that the BRD is not part of release deliverables; however, it is a key activity and deliverable of the project process. A support and escalation document is not considered an added value to the project process, however it is essential to the release process since the release process is concerned with the quality operational delivery of the service through retirement.

Release Plan and Project Plan

When release management is being implemented and the introduction of a release schedule is introduced, more confusion arises. Why do we need two plans to manage one project? The confusion can be compounded as the release plan feeds the project plan and the project plan relates to the release plan; so why have both? The answer lies with the same concepts that have already been discussed; the two are complementary of each other and each has their own goals and objectives.

The project plan will be created by the project manager after the scope of the project has been approved. The plan will contain the activities that need to be completed, what resources are needed to complete the activities, milestones, and a timeline for delivery. Once these activities are completed and delivered, the project is completed and the project plan is closed. The release plan is completed and monitored by the release project manager once the initial project activities such as scoping, estimation, and approval have been completed. Milestones on the release plan are associated with monitoring the quality of the release being created and ensuring the project will be able to operate once it is introduced into the production environment. Checkpoints of the release plan will include review of activities such as the infrastructure once it is built and QA testing to ensure any bugs that have been discovered are documented, corrected, or mitigated prior to implementation and operational readiness testing. Another activity that is not usually part of the project plan is the planning for early life support of the service once it is introduced. While the project is not typically concerned with this, release is concerned since it is focused on the total cost of operation of the service.

Tying It Together

Release management and project management do possess similarities and are often confused. The confusion is compounded by not understanding the goals, focus, and objectives of each of the processes. Both processes have an important role in the development of business and IT services. These roles and activities are

© 2010 by Taylor and Francis Group, LLC

complementary and if blended successfully, can produce quality services that are cost effective, reliable, maintainable, and deliver value to the customer. The customer experiences increased uptime and quicker return to service in the event of an incident.

The simple difference between the two processes is that project management has a start and an end, and release management is concerned with the overall lifecycle of the service. Figure 3.1, the production/service lifecycle, illustrates this difference. The inception of a service begins when an organization identifies a business need or problem and creates a business service to fill the need. Many times, especially in today's world, IT is asked to create IT services to support the business service. This is when the initial release of the business or IT service is created. To create the first release, a project is created, approved, funded, and built as illustrated by Project 1 in Figure 3.1. The project has a start and an end. The service continues to operate; it is then decided that improvements need to be made to the service, and this is when subsequent releases are created. However, unless the subsequent releases are scheduled using a release schedule, planned improvements may not be funded and delivered on time. The quality of the service starts to degrade, incidents occur, and productivity of the organization decreases. Creating a release schedule ensures the new or updated releases are scheduled and implemented, always improving the service capabilities and performance.

Figure 3.1 ties together the concepts of release management that have been discussed in this chapter and in Chapter 2. Creating a release practice entails installing all the various components that have been discussed and are illustrated. One important aspect to keep in mind is that while all of the components should be installed, they do not need to be fully matured to have a successful release practice. The pieces that will bring the most value to the organization should be installed first. Also, maturing the release practice is a continual process and maturity of the practice will develop as the organization matures. An important first step in developing a release practice is creating a strategy.

Once the decision is made to embark on the implementation of a release, process-specific objectives and goals must be established to ensure that there is focus and direction in the development and implementation. Setting objectives and goals should follow the SMART approach—specific, measurable, attainable, realistic, and timely. By using this approach the developmental process will achieve focus and be successful.

In Chapter 2 a simple strategy model was discussed. Using this model will provide an excellent vehicle to direct the objective and goal setting for the release process.

- **Where do we want to be?** What problem are we trying to solve? What is the organization going to look like after the release implementation? What benefits are going to be realized? How will the release practice be positioned in the organization? How will the project management process be positioned? How will the release process align with project management?

© 2010 by Taylor and Francis Group, LLC

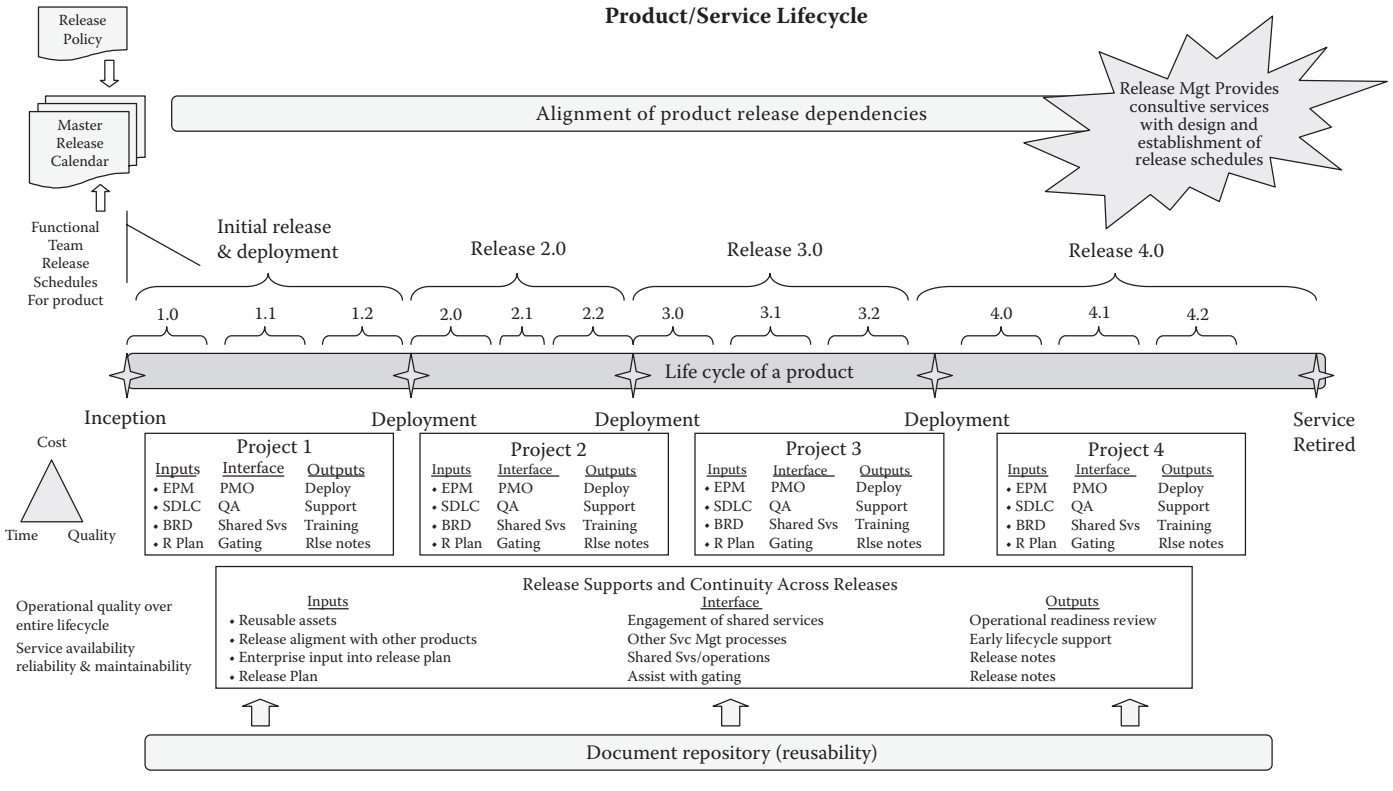


Figure 3.1 Product/service lifecycle.

© 2010 by Taylor and Francis Group, LLC

- **Where are we now?** This is when the baseline of the existing process or lack of process is established. Are there any release activities currently being executed? If so, do they currently align with the project management process or are the processes in conflict? This is an important step that many organizations overlook or bypass simply because there is limited or no existing process. However, even if there is no existing process, the "as is" status needs to be documented, so that the organization will be able to measure the success that is achieved when the process is implemented.
- **How are we going to get there?** Using the strategy and objectives created in the first step will provide the guidance to develop the implementation requirements. The requirements can then be translated into an implementation roadmap. Using the roadmap will provide the plan for a successful implementation that maps back the strategy and objectives.
- **How do we know we got there?** Using the established goals, objectives, and roadmap, the activities achieved during implementation can be measured to determine when each goal and objective has been achieved. An important aspect of process development is to establish metrics and measurements that will provide transparency of the performance of the process. Without metrics and measurements, there is not any way to demonstrate the success and value that release management brings to the organization. In a later chapter the establishment of, and method of using, metrics will be covered.

Software/System Development Lifecycle (SDLC) vs. the Release Lifecycle (RLC)

Once the strategy is created and it has been determined how that strategy will be realized by release management and project management, the next step is to understand how to execute the strategy. Generally in organizations when a project is approved and started, some type of project methodology is employed, whether it is a waterfall approach or an iterative approach. This project methodology is a roadmap of the different phases that need be completed to deliver a project. The release process should also use some type of lifecycle or roadmap. In earlier chapters the release lifecycle (RLC) was introduced. The RLC is the roadmap that is used for release management to ensure quality delivery of releases. The RLC is created to align with a software development lifecycle (SDLC) to gain the most benefit from both processes.

Both the RLC and the SDLC have specific goals and objectives, as previously discussed. Figure 3.2 illustrates the alignment between the two. While it appears many of the segments are similar, each has a different focus. In a later chapter the detail and deliverables of the release lifecycle will be discussed.

When many organizations start to look at release management, the traditional thinking is that it either replaces the SDLC or is only applicable to software

© 2010 by Taylor and Francis Group, LLC

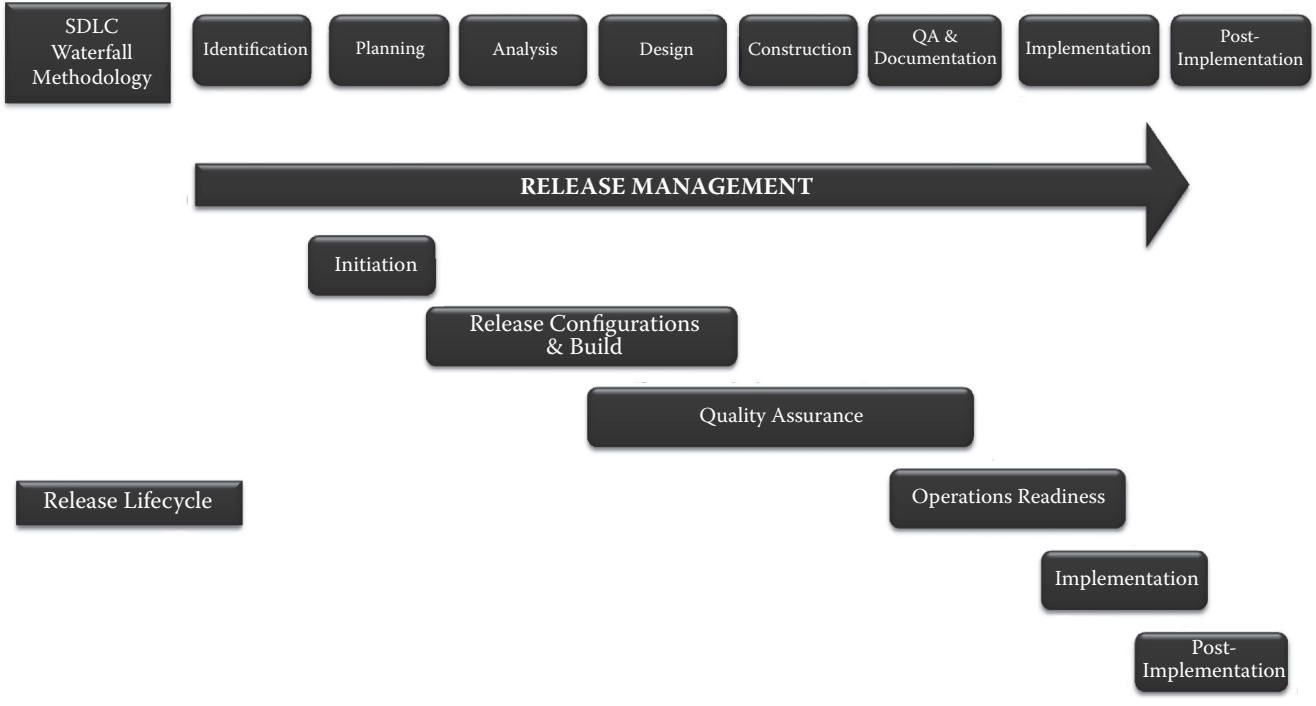


Figure 3.2 Alignment of RLC and SDLC.

© 2010 by Taylor and Francis Group, LLC

development. Using a release process only for software development is one approach; however, to fully gain the benefit of a release process, the process must be applied to the entire release development engagement. In the RLC, which is illustrated in Figure 3.2, there is a step called "release configuration and build." This step includes the development and building of the infrastructure pieces of the service being created. Whether it is the creation of a new environment, network, or database, the development and testing of these components of the release must be included in the RLC. Without ensuring that these pieces of the release have been tested and meet the quality standards set forth, the overall performance and quality of the release may be compromised. This step will be covered in more depth in a later chapter.

Alignment Pyramid

Thus far this chapter has focused on trying to understand the differences and similarities between release management and project management. There has been some discussion that the complementary processes need to be aligned and how this alignment will benefit the organization.

The Alignment Pyramid in Figure 3.3 illustrates the different phases needed to achieve the benefit from the synergy of using both release management and project management. Like any other pyramid, the different levels depend on the soundness of the preceding level to build the strength of the entire pyramid. Creating a strong base determines the strength of the pinnacle. Figure 3.3 also calls out high-level concepts that both release management and project management have to accept to create a strong level.

- **Alignment**—Creating a strong base is the key to the entire alignment. If alignment cannot be agreed upon, then the pinnacle of quality desired will be tough to reach. To gain alignment, project management must take on the values of the release process and a longer-term view versus the short-term project end view. Release management must also take on the values of the project management process, such as understanding the short-term view.
- **Benefits**—Once a strong base of alignment is achieved, benefits can start to be realized. The benefits are mainly process focused. Project management gains the benefit of reusability of components, and better forward scheduling results in better use of resources and improves regulatory compliance. Release management gains a benefit from the discipline of the project management processes and the deliverables that result.
- **Operational**—As the alignment continues to strengthen, the benefits start to translate into stronger operational outcomes such as decreased time to delivery and improved availability, reliability, and maintainability of the service being created and released into the production environment.

© 2010 by Taylor and Francis Group, LLC

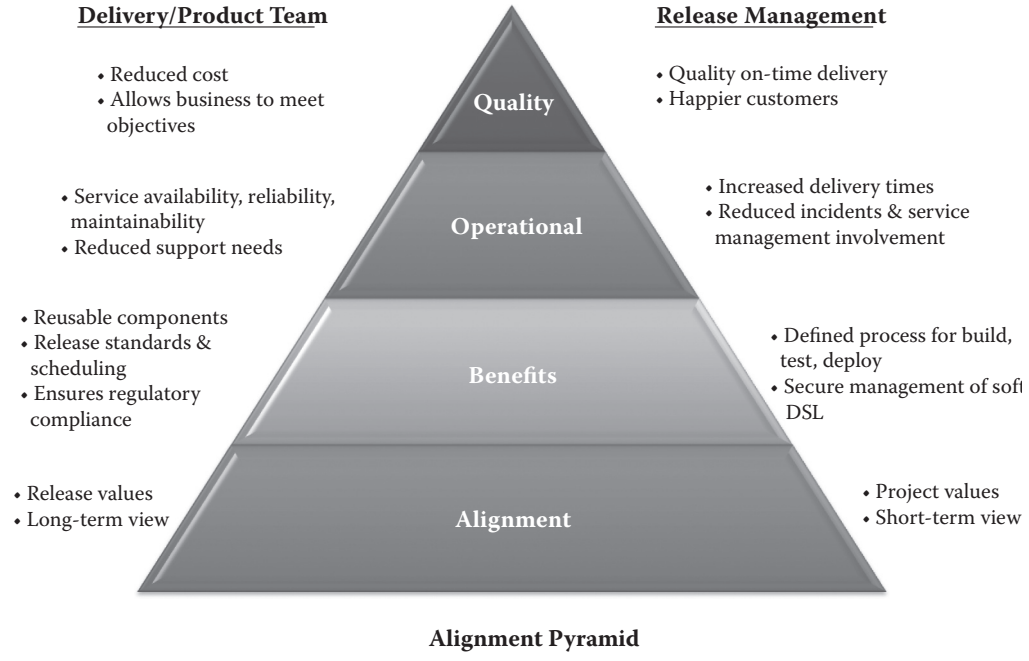


Figure 3.3 Alignment pyramid.

© 2010 by Taylor and Francis Group, LLC

- **Quality**—The pinnacle of the pyramid is where the release process and the project process are in alignment. The results of this alignment include reduced cost of the delivery of the release, quality on-time delivery, and the business's ability to achieve their goals and objectives. These benefits result in happier customers.

Concept Review

Release and project management are commonly confused as the same process with the same deliverables. Not understanding the differences and similarities between the two processes causes many organizations to either not implement release management processes or to try to merge release management principals into the project management process. This chapter has discussed the similarities and differences between the processes, and how the two processes can complement each other. The concepts reviewed were:

- There are differences between release management and project management. These differences lie in the objective, focus, and goals of each of the processes.
- While there are differences, there are similarities between the two processes. These similarities can be used to strengthen each process through the sharing of tasks and deliverables.
- While some deliverables may be exclusive to one process, they can be complementary to the other. Once these complementary processes are aligned, the quality of the service being delivered increases.
- The differences, similarities, and roles of the release plan and project plan were reviewed.
- The product/service lifecycle was reviewed including the role release management and project management play in the lifecycle of the service.
- Creating a strategy for implementing release and project management will guide the organizational goals and objectives and will help with the alignment between the two processes. The SMART concept was introduced in strategy creation as well as a simple strategy model.
- Using the release lifecycle and service/system development lifecycle in alignment will improve the development and execution of the service. The release lifecycle is not just for application development; it also includes the development of infrastructure components.
- The Alignment Pyramid illustrates how the alignment of release management and project management can result in better-quality services, reduced development cost, and happier customers.

© 2010 by Taylor and Francis Group, LLC

Chapter 4

Release Management Is Not Just for Software Development

Today, more and more organizations are embracing Information Technology Infrastructure Library (ITIL) and service management frameworks. One of the challenges is that many of these organizations do not gain the full benefit of these frameworks. These organizations look at service management as an infrastructure initiative and do not understand that it involves the entire lifecycle of service delivery. Yes, the service management process does involve the development and operation of the organization's infrastructure operation, but it also includes the incident, problem, and service level processes. Service management also includes the development and release of services into the IT environment. On the surface, most organizations would agree that some type of development process should be followed in the development of infrastructure components, but in reality, if there is an infrastructure development process, it is done in isolation within the infrastructure management group and not as part of the holistic service development process. Therefore, the benefit derived from a lifecycle approach is not realized.

Holistic Approach

In order for an organization to realize the full benefit that can be derived from a good release process and the service management lifecycle, a holistic approach

© 2010 by Taylor and Francis Group, LLC

toward the entire development activity must be understood and utilized. A holistic approach simply ensures that all facets of the design, build, testing, acceptance, implementation, and operation of a service relate to each other. They are linked by a common strategy that should be traceable back to the organizational strategy and mission. The release process should be allowed to take the holistic approach to the delivery of the service, which means the release process has touchpoints in software development *and* in the infrastructure component development.

Being able to relate the strategy of the service and the release of the service being created to the building of the requisite infrastructure components increases the fit for purpose and fit for use of those components. Ensuring the infrastructure components are right sized eliminates overbuilding and reduces unnecessary cost. Creating release processes that are inclusive of the infrastructure increases the quality of the release and ensures maximum operational stability, maintainability, and reliability.

Release Management and the Infrastructure

Earlier, the idea that many organizations perceive that service management is only related to the infrastructure was raised, and that to gain the most value, a holistic approach to delivering service should be used. The idea that service management is related only to infrastructure is manifested by the direct interaction with change and configuration management, and the idea that infrastructure is related to any type of development process is sometimes a foreign idea. Therefore, when the idea of release management being involved with the building and provisioning of infrastructure components is raised, the common reaction is to ask why. There is also denial and obstinacy to allowing release processes to be involved. These reactions are normal when there is a lack of understanding. Release management's involvement in infrastructure is not one of doing and building, it is one of planning, reviewing, and oversight. In other words, release plays the same role in infrastructure development as it does in software development—ensuring that all parts of the release are developed to meet requirements, tested for quality, reviewed for operational readiness, and implemented.

New Environment Request Process

The next chapter deals with the release lifecycle (RLC) and will go into great depth in explaining the different phases of the lifecycle. One of the phases within the RLC has more of a direct relationship to infrastructure than the others; this phase is called release configuration and build. Within this phase there is a subprocess called the new environment request (NER) process. The NER process deals directly with the planning, building, configuring, testing, and implementing of the environment that will support the software being built or modified within the

© 2010 by Taylor and Francis Group, LLC

approved release. The title of the process, new environment request, may imply that the process only needs to be executed if new hardware to support the release needs to be procured. On the contrary, the NER process needs to be executed if there is not already an existing environment to run the software. The NER process provides a defined method to determine what type of environment is needed to run the service; this includes understanding capacity and availability requirements as well as the needed infrastructure components to meet these requirements. Once this information is known, the architecture group can determine how to best provide the environment required. This may mean utilizing existing server and database capacity that is not being fully utilized, using a virtualization approach, or if no other existing resources are available, procuring the needed infrastructure components. While release management does not provide the inputs to the process, it does provide the oversight and guidance for executing the process.

The process diagram illustrated in Figure 4.1 depicts the high-level process flow of the NER process. The NER process involves many areas within the IT department as well as minor involvement from the customer requesting the service. The inputs into the NER process come from the customer, project management, network engineering, design and engineering, IT architecture, application teams, as well as other IT groups such as database administrators and engineers. Release management acts as the guide and adviser to help development teams navigate through the process and ensure that the release stays on track.

The NER process may not be well-received by the organization simply because the process asks that a structured planning and documentation process take place. However, all of the activities taking place within the process are activities that are normally done when building a new environment. Using a structured approach

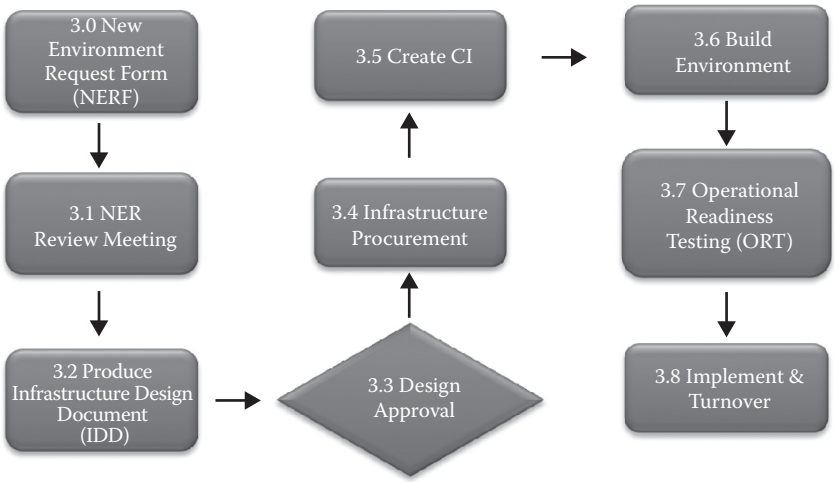


Figure 4.1 New environment request (NER) process.

© 2010 by Taylor and Francis Group, LLC

will improve the quality of the environment and will actually reduce the time to delivery. As a point of reference, the time to deliver a new environment from conception or request to delivery can be from four to twenty weeks or more depending on whether equipment needs to be procured and on the efficiency of the organization. Utilization of the NER process can typically deliver an environment within four to eight weeks.

One of the premises of this book is to provide suggested approaches that can be used to create a release management practice within an organization. The processes and concepts described in this book can be utilized with minimal customization; the customizations needed should be made to fit the culture of the organization. The processes and concepts are plug-and-play solutions, and it should be remembered that flexibility is key in the execution. In keeping with the "how to" spirit and providing usable approaches, the NER process will be the first subprocess to be discussed in depth. It is hoped that this in-depth approach will shorten the time of process implementation.

This chapter is focused on how release management interfaces with infrastructure development, and the NER process is central to this focus. Even though the NER process is not the beginning of the release process, discussion of the NER at this point is in alignment with the focus of this chapter, and a detailed description of the NER process therefore appears below. In keeping with the placement of NER in the release lifecycle, the numbering schema begins with 3.0 in Figure 4.1.

NER Process Description

NER Entrance. As with any process, there has to be a beginning or an entrance into the process. To increase the chances of a successful outcome of the process, specific requirements and criteria must be met prior to entering the process. Creating and publishing consistent and repeatable entrance criteria will not only increase the success of the release, it will also reduce the development time and cost of the release. The following are suggested entrance criteria to the NER process area:

- A business requirements document (BRD) should be created that provides an understanding of the requirements and functionality of the release.
- Release and project plans need to be completed to understand the release implementation dates and timelines.
- Availability and capacity requirements must be considered.
- Continuity requirements must be examined based on business criticality.
- Budget considerations must be addressed.
- Any infrastructure requirements set forth in vendor-supported software must be addressed.
- Types of environment(s) needed must be considered; for example, development, testing, training, staging, and production. As a point of reference, each environment may need a separate design document.

© 2010 by Taylor and Francis Group, LLC

Step 3.0 New Environment Request Form (NERF)

Defining the prerequisite criteria for entrance into the NER process is an important step, and so is capturing the critical requirements information and data so that it can be used within the process. The information and data can be documented by using a template that captures the required data. Utilization of a template ensures the necessary requirements and data are collected consistently, reduces rework because the correct data is gathered, and reduces the development time. Therefore, the new environment request form (NERF) is the artifact that needs to be completed and approved to enter the process.

The NERF should be completed by the project manager in conjunction with the systems architect and business systems analyst (BSA). It may appear that not all of the necessary technical groups that will be involved with the creation of the environment are being consulted, but at this point in the process, these groups do not need to be involved. The NERF's primary focus is to capture the information and requirements of the release that will be used later in the process to design and build the environment.

The three resources mentioned—the project manager, systems architect, and the BSA—each have a role in completing the NERF and each brings a different perspective to the completion of the template.

- **Project manager (PM)**—has overall responsibility for gathering needed data and completion of the form. The PM will be able to supply project-specific information, such as budgetary information, timeline information, and release resource information. The PM will also supply the release implantation information.
- **Business systems analyst (BSA)**—will provide availability, capacity, and the recovery requirements captured from the customer and documented in the business requirements document. The BSA will also provide the project summary of the release focusing on the problem that is being solved and the expected outcome of the release.
- **Systems architect**—provides the technical input for the NERF, translating the availability, capacity, and recovery requirements into initial environment requirements; for example, what type of servers should be used. The systems architect also will be responsible for the completion of a preliminary infrastructure impact assessment.

The NERF is a key artifact within the NER process; it provides the pertinent information to begin the design of the environment. The requirements captured in the NERF are categorized into key areas.

- Project summary
- Budgetary information

© 2010 by Taylor and Francis Group, LLC

- High-level scope
- Type of environment
- Capacity and availability requirements
- Infrastructure impact assessment
- Implementation information

A generic NERF template has been included in Appendix E. This template can be used in its present form, however it is recommended that it be customized for the organization using the form. Customizations may include roles of resources, job titles, and department names, just to name a few. An important concept to remember when completing the NERF is that it does not need to be completed in a sequential manner; in fact, in most cases sequential completion will be difficult because inputs from various sources are needed. Once the NERF has been completed it will be used in the NER review meeting to gain approval to move forward.

Step 3.1 NER Review Meeting

Once the NERF has been completed it is necessary to obtain approval to move forward with the design and building of the new environment. Depending on the specific organization, approval can be handled in different ways. Some organizations may have an infrastructure standards committee that would review and approve the NERF, or it may be a single individual such as the chief technology officer (CTO) or an infrastructure director. The idea of this approval phase is to ensure the requested architecture and environment meets the organization's standards and the approvals have been given prior to moving forward. These approvals are both from a resource allocation perspective and a financial perspective; in other words, is there an approval to spend the money.

The release manager plays the role of facilitator in the NER review meeting. Once the project manager has the completed NERF it should be submitted to the release manager. The release manager reviews it for completion and ensures that all the required information is contained within the form. The release manager does not review the contents for correctness of approach or compliance with standards; that is the role of the approving entity. Once reviewed, the release manger will schedule the review meeting with the approving body where the NERF will be reviewed and approved. Depending on the volume of requests, a standing approval meeting could be scheduled whether it is weekly or monthly.

Step 3.2 Produce the Infrastructure Design Document

Now that the approval has been received, the next step is to create the detailed design of the requested environment. This step may be one of the more lengthy steps within the NER process due to the complexity and number of functions that

© 2010 by Taylor and Francis Group, LLC

need to be involved. However, if the resources have been allocated to this activity, the time can be reduced significantly. Another consideration involves the components needed in the requested environment.

In today's world of virtualization, service-oriented architecture (SOA), and cloud computing, the resources that must be involved in designing an environment can vary significantly depending on how the environment is going to be used to deliver the service being created. The systems architect assigned to the development of the environment will take the lead in this activity, although it is also important for the project manager to stay connected to ensure that the process is moving forward and will meet the milestones.

Generally, the contributing infrastructure groups that need to be engaged in developing the detailed design will be server design and engineering, enterprise storage, network design, and database design groups. Depending on the organization, these technology groups may have different names. One group that must be included and is sometimes overlooked is the enterprise security group, which will ensure the environment being designed meets security standards; it is important to design for security. All of these various groups will contribute to the architecture being designed.

Capturing the input from these groups can be done in various ways. The inputs can be captured and documented in one document or each group's input can be documented in separate documents. If captured in separate documents, a document named the infrastructure design document (IDD) is used as a supporting document for the various groups that have submitted independent documentation. A generic IDD can be found in Appendix F along with security risk assessment, new database request, storage request, and firewall access request forms. Other documents required are system diagrams that illustrate the entire environment as well as any network diagrams.

The goal of this subprocess is to present a detailed architectural design that can be used to build and deliver an environment to the project team. This process is important because it is iterative since most systems will require a development, test, training, stage, and production environment. While some project groups will request that all environments be built at once, it is advisable to build the development environment first to determine if the environment meets the requirements of the service. Oftentimes, if the requirements are not clear, there will be changes to the configuration of the environment.

Step 3.3 Design Approval

You may notice that so far in the process there have not been any activities to procure or build the requested environment. No resources have been utilized to create and document the design, and there have been no activities that could be deemed as wasteful. Creating a well-thought-out design reduces the amount of rework and associated cost.

© 2010 by Taylor and Francis Group, LLC

The design approval activity is similar to the NER review meeting. Once the detailed design is completed and documented by the system architect, the project manager who has been working in conjunction with the system architect provides the IDD and supporting documents to the release manager. The release manager reviews the submitted IDD to ensure that all of the required information is contained within the form; the release manager does not review the contents for correctness of approach or compliance with standards. Once it is reviewed and considered complete, the release manager will schedule a meeting with the approving entity for review and approval. If the design meets the organization's standards, is a sound architecture that will be able to deliver the availability, stability, maintainability, and usability requirements, and meets security standards, the design is approved and work begins. If not, the design is rejected and sent back to the system architect for remediation.

Step 3.4 Infrastructure Procurement

Procurement is not always necessary in providing a new environment. The organization may already have unused equipment or equipment that can be reprovisioned. Additionally, a virtualized solution may have been designed. If that is the situation, this step in the process can be skipped. However, if equipment needs to be procured, it is advisable to place the order for the equipment as soon as requirements are known and approved. Procurement can lengthen the time for delivery of the new environment to more that sixteen weeks; the organization is at the mercy of the equipment manufacturer.

The procurement process for each organization is unique in every organization, and therefore the actual process of procurement will not be discussed here. It is recommended that you follow the prescribed process within the organization, and proper planning can help to expedite the process.

Step 3.5 Create the Configuration Items

Many organizations have some type of configuration management database (CMDB) whether it is a mature enterprisewide tool containing the entire organization's Configuration Items (CIs) including their relationships to each other and the business services they support, or as simple as a spreadsheet that lists infrastructure components with a few attributes. Whichever CMDB solution the organization is utilizing, it is important to record any new components or CIs introduced into the environment. There may be a question of why this subprocess exists this early in the process before the environment is built. It is a matter of being able to have an accurate accounting of all CIs in the environment and their status. When a new infrastructure component is ordered, it should be created in the CMDB. It is acceptable to wait until the component is received; however, it needs to be noted in

© 2010 by Taylor and Francis Group, LLC

the CMDB when it is received. You will not be able to add all of the attributes of the CI at this time; as the CI is inserted into the architecture, more attributes and relationships can be populated for the CI. In a future step of the NER process the final attributes and relationships of the CI will be populated.

The process of creating and populating the attributes needs to follow the configuration process established by the organization.

Step 3.6 Build Environment

The build environment subprocess in the NER process can be variable due to the differences in the types of environments being built. The commonality is that the time it takes to build, test, and implement an environment will be reduced if there has been proper planning; the quality of the build will also be improved and the cost will be reduced. Using the IDD and other design documents, the engineers will have a documented roadmap of what needs to be done. Build procedures will be technology and organization specific and as with any process, there should always be a continual improvement mechanism in place to improve the processes.

Step 3.7 Operations Readiness Testing

The environment has been built and unit tested to ensure the new environment will meet the requirements set forth in the IDD. A review of the test results is conducted with the project team to demonstrate and confirm this. Once acceptance is gained, the environment will be readied for turnover to the project team. This is done through a process called operational readiness testing (ORT). Depending on the organization, an environment or service may go through one or more ORTs prior to implementation. It is recommended at least two ORTs be done prior to implementation; the primary ORT is conducted after the initial environment build; the second ORT will be conducted prior to implementation. Each of the ORTs will have similar testing, although the second will be somewhat more extensive as this is the final technical verification to ensure the environment is functioning as designed. The second ORT will be reviewed in the next chapter.

ORT has two primary objectives. The first objective is to ensure that the end-to-end functionality of the environment is functioning as designed. The second objective is to ensure that the operational components have been installed and are functioning. Operational components can be system- or functionality-monitoring tools and virus-prevention software. Additionally, a review of the installed operating system is done to ensure the current version is up to date. It is possible that an update or patch may have been issued since the operating system image being used was created. Another aspect to be considered during the ORT is whether security standards have been built into the environment; if not, this could expose the organization to additional risk.

© 2010 by Taylor and Francis Group, LLC

The final step of the ORT process is to update the CI's status and attributes that have been tested in the CMDB. It is important the status of the CIs and services are updated to reflect the current status.

Step 3.8 Implement and Turnover

The final step of the NER process is the implementation and turnover of the environment to the project team so the development of the service or release can be continued. In the ORT process it was mentioned that the project team will review the environment that was built; this is a preliminary review prior to ORT. Once the ORT has been completed, the environment is moved out of build status and is recognized as an implemented CI. This doesn't mean it was implemented into the production environment, it means it is now within the scope and control of configuration management and within operational control. Being within operational control means the environment is being monitored to ensure it remains functional and operating as designed. If the environment becomes unavailable, steps are taken to restore service of the environment; after all, if the environment is not operational, development teams cannot develop the service.

The final step in turning over the environment is to gain sign-off from the project team. It is recommended that a formal turnover agreement be signed. Depending on the makeup of the organization, this final sign-off may have contractual and financial implications. Some organizations may outsource their infrastructure services and therefore the delivery of the service would be a contractual milestone and generate payment. If there are contractual considerations, a warranty period or other considerations need to be created.

Concept Review

Release management is not just for software development. This chapter illustrates the concept of release management taking a holistic approach to system and service development, and being holistic means including infrastructure development.

- The organization will gain the biggest benefit of release management when a holistic approach is taken and consideration of how the development of infrastructure that supports the software being developed is interwoven.
- A holistic approach ensures all facets of the release development and build are connected and aligned to the mission and strategy of the organization.
- When release management is inserted into the infrastructure design and build process, there may be pushback from infrastructure teams due to a lack of understanding of the holistic connections and strategy alignment.

© 2010 by Taylor and Francis Group, LLC

- Relating the strategy of the service being created to the building of the requisite infrastructure components increases the fit for purpose and fit for use of those components.
- The new environment request process was introduced and the steps of the process were explained.
- The role of release management in the NER process is one of review and oversight, not one of creating the design or building the environment.
- The introduced NER process is a generic process and should be tailored to the individual organization.

© 2010 by Taylor and Francis Group, LLC

Chapter 5

The Release Lifecycle

One of the principal motivations for this book is to share practices that translate the concepts and theories of release management into practical activities that can jumpstart a release practice. So far the basic concepts of release management have been discussed; these concepts form a basis for the use of the release lifecycle (RLC). This chapter will focus on the RLC and will go into detail with respect to the different steps and activities. During the RLC, various artifacts will be used to illustrate the types of activities and information that need to be captured, documented, and distributed. Generic artifacts that can be used to execute a RLC are included in this book. The artifacts can also be customized to fit any organization.

Benefits

A lifecycle is defined as a continuous sequence of changes and a series of stages to reach a desired state. Using a lifecycle for delivering a release process is a natural fit. After all, release management is a continuous iteration of changes to a service or product from its inception to its retirement. If these changes are not controlled, aligned, and quality assured, the integrity of the service or product degrades and does not meet the business customer's needs and expectations. Some organizations attempt to implement a release practice without an organized process or lifecycle. They feel that simply implementing a release schedule is release management and do not venture into the areas of guidance, facilitation, and governance that deliver the greatest benefit. These benefits are added through the use of the RLC.

The benefits of the RLC are related to efficiency, quality, and financial issues, and all three of the benefits are realized through consistent, repeatable, and sustainable processes.

© 2010 by Taylor and Francis Group, LLC

- **Consistency**—The idea of consistency is always recognized as a desirable attribute, however it is also very difficult to realize. The RLC is a defined process with specific steps and specific tasks that need to be completed. Consistency comes with the requirement that defined activities need to be completed prior to implementing the release into production. Understanding by the development team of the consistent activities required drives efficiency within the development process by eliminating misunderstanding of what needs to be completed. This understanding also increases the quality of the release because there is a defined list of value-added activities that are consistent for every release. Eliminating rework through the clarity of consistent activities reduces the cost of development.
- **Repeatability**—Similar to consistency of the task and activities, having a defined lifecycle increases the repeatability of the task being performed. There is a saying that practice makes perfect, and this applies to the RLC. As development teams become more familiar with the tasks of the RLC through repetition, the quality of the release will increase. At the same time, as the teams repeat the tasks for more and more releases, the way the tasks are completed will become more efficient. When the teams become more efficient, the time spent doing the tasks will decrease, reducing the cost of resource's time.
- **Sustainability**—One of the most difficult aspects of implementing any process is sustainability. It is not uncommon for organizations to have a process implemented and for it to degrade to a point that the implemented process is no longer in practice. Another common occurrence is for an organization to implement a process and not let it become stable before the organization tries to improve it. Sustainability and longevity of a process occurs when real benefit and value are realized. Efficiency benefits are realized through consistent and repeatable tasks, however efficiencies increase significantly as resources continue to use the RLC and the process becomes part of the day-to-day activities. This also occurs with respect to increasing the quality of the release. The financial benefits will continue to multiply the longer the RLC is sustained.

Implementation Considerations

The release lifecycle has been created as a way to illustrate and define activities and tasks to facilitate a quality release that is reliable, maintainable, and serviceable. The release will meet the requirements of the customer and enables the business to achieve its goals and objectives. While there may be other approaches to achieving the goals of a release process, the creation of a RLC has been found to be an efficient method for resources to understand and navigate the various activities, resulting in a quality release.

There are basic concepts that must be remembered when implementing the RLC to make the implementation successful. First, do not create a process that is

© 2010 by Taylor and Francis Group, LLC

overly burdensome. When the RLC is first implemented, many resources will see it as overhead and not understand why certain activities and tasks within the RLC need to be completed. Effort should be made to only create activities and tasks that are value added. When the activities within the RLC are developed, the question should always be asked, "What value does this activity bring to the release?"

Secondly, be able to articulate the goals and objectives that the RLC brings the process—what is the value added? Any resource involved in either the creation or the operational delivery of the RLC needs to be able to describe the goals, benefits, and objectives.

Third, the RLC should be designed to fit the organization. This is not to say that the focus and objectives should be redirected away from activities and tasks that promote the delivery of a quality release. On the contrary, the core values of release management need to be maintained, however the processes should not be inflexible and foreign to the organization. Use the culture of the organization to be successful; ask yourself, "How have other processes been successfully implemented into the organization?" This topic will be covered in more depth in another chapter.

The Release Lifecycle

The release lifecycle (see Figure 5.1) is a process within the release management practice that serves as a roadmap for delivering a quality release. Each of the phases is related to the others and some are implemented simultaneously. The RLC includes seven phases:

- Initiation
- New Environment Request (NER)
- Quality Assurance (QA)
- Release configuration and build
- Operational Readiness (OR)
- Implementation
- Post-implementation

Each of the phases has a defined purpose and benefit, and each phase requires specific deliverables designed to ensure that the release is properly designed, constructed, documented, tested, and operationally ready prior to implementation. At the conclusion of each phase of the RLC, a review of the artifacts and deliverables is completed by the release manager to ensure the release remains on schedule and is meeting the necessary quality standards.

It is important to note that the RLC process does not always follow a linear flow, and that some of the process activities run parallel to others; this is illustrated in the phase process flow. For example, the activities in the release configuration and build phase occur at the same time that the environment is being built within the NER

© 2010 by Taylor and Francis Group, LLC

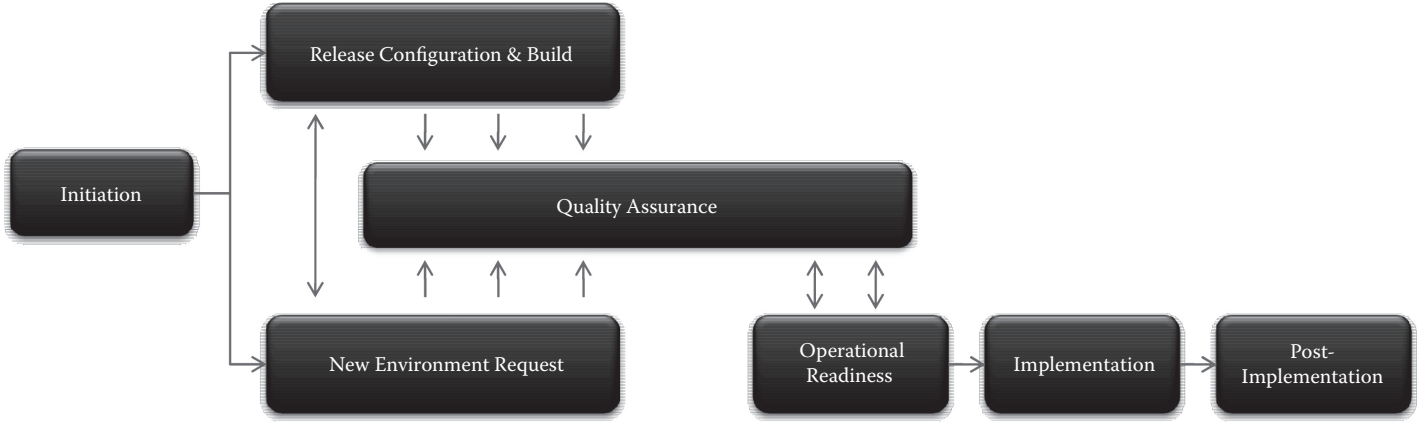


Figure 5.1 Release lifecycle.

© 2010 by Taylor and Francis Group, LLC

phase. Additionally, QA testing activities are ongoing at various points throughout the RLC, not just during one designated period in the development cycle.

The release process will be utilized for all releases from large, enterprisewide implementations to small minor releases. This is not to say that every task within the different phases should be used for every release, however the objectives of each phase should be met. The extent of the artifacts and deliverables that will be required will be determined during the Initiation phase.

Artifacts

Artifacts and templates utilized throughout the RLC should be used with an enabling philosophy in mind and should add value to the process. They should be designed and used to provide a means for documenting the tasks and activities of the release management process. The documents provide a consistent and repeatable means for reviewing and ensuring that the activities and tasks being completed will result in a quality, operational-ready release that meets the requirements of the business. The artifacts also provide evidence of compliance with the governance aspect of release management.

Artifacts are classified into three categories:

- **Informational**—used to provide general information about the release and as a basis for knowledge and understanding. These artifacts do not require approval or acknowledgment of review.
- **Review**—information provided in these artifacts requires review and specific understanding. There is an acknowledgment section within the artifact indicating this.
- **Approval**—these artifacts require written approval from the designed approval authority. These artifacts are used for verification of compliance with regulations such as the Sarbanes–Oxley Act (SOX) and other governmental requirements. They also are used to provide approvals within the process such as approval for procurement.

Another category of documents and artifacts utilized throughout the RLC is either control or support documents.

- **Control**—documents that are designated for demonstration of compliance.
- **Support**—documents used to provide the necessary information to support the release in operation.

Each of the phases of the RLC will have a mixture of control and support documents; each phase will contain at least one control document. The documents and artifacts that are illustrated in the RLC within this book have proven to be effective, but it should be kept in mind that the documents and artifacts used should

© 2010 by Taylor and Francis Group, LLC

fit the organization that is using them. There is no set number of documents; if more or less are required to be effective within the organization, then that is what should be used as long as they are providing value. In fact, the sample documents presented in this book are not all inclusive of the documents and artifacts needed to successfully implement a release. Within the description of each phase, samples of some artifacts are included while others mentioned are not. The artifacts included are specific to the release process while the others are used primarily by other developmental processes, for example, the business requirements document (BRD) is a project management artifact used by the release process.

The sample artifacts contained in this book are just samples. They can be used as presented, however it is advisable to review the artifacts and customize the content to fit the specific organization.

Initiation Phase

Initiation is the entrance phase into the release lifecycle and begins after the release has been approved. The type of approval process will vary from organization to organization. Typically the approval process will include some type of a concept proposal—a business case to support the need for the release—and may include a feasibility analysis to determine an estimate of effort and cost. While these types of artifacts are not deliverables of the release process, they do play an important role in building the release and should be available to the release manager. These artifacts will provide the basis of understanding of the release and will enable the release manager to provide guidance and governance throughout the RLC.

Once a release is approved, a release project manager (RPM) or project manager (PM) is assigned to the release. The PM will begin to work with the project team to capture and document the business requirements, create the functional requirements, and create a project charter. Once these artifacts are complete, this is the trigger to enter into the first phase of the RLC—Initiation.

The focus of the Initiation phase is to capture the strategy and provide a basis for the building and configuration of a quality, operational-ready release. This phase will use several artifacts created within the approval and project manager processes. These documents, which were previously mentioned, provide the basis for subsequent artifacts that will be used within the RLC (see Figure 5.2).

Step 1.0 Initiation

RLC initiation begins after the approval of the release and the creation of artifacts that provide the strategy and overview of the release. The artifacts listed in the previous section need to be a prerequisite for the entrance into the RLC; without them,

© 2010 by Taylor and Francis Group, LLC

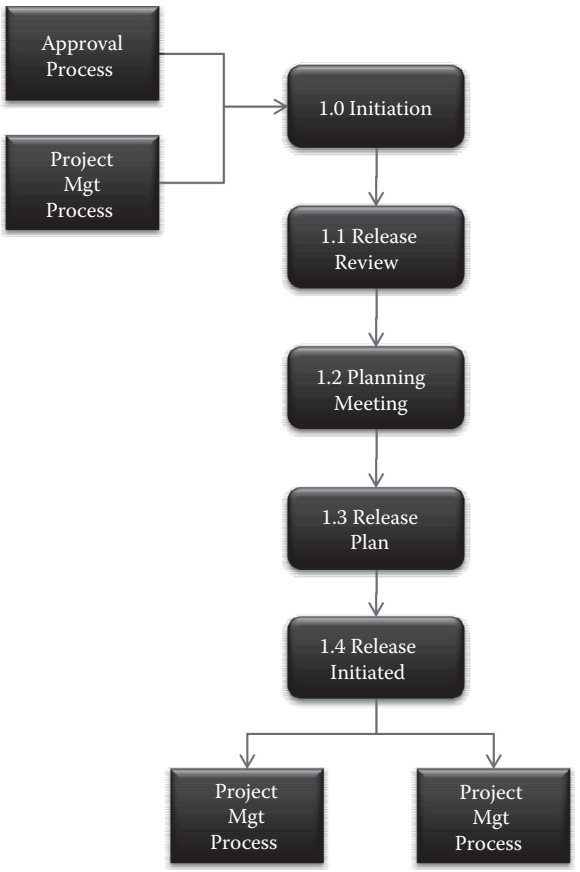


Figure 5.2 Initiation phase.

the chances for a successful release are very marginal. It is recommended to create and incorporate a list of the prerequisite artifacts and include them on the release deliverables checklist, which will ensure the critical data is captured. The release deliverables checklist has not been mentioned previously. It is a checklist used by the RPM and release management (RM) as a roadmap of artifacts and deliverables needed to complete the release. The usage of the release checklist will be reviewed in upcoming steps within the process.

Step 1.1 Release Review

An effective release manager needs to have a full understanding of the goals, objectives, and strategy of the release being created. Without this understanding, the RM cannot effectively provide the guidance and oversight needed to

© 2010 by Taylor and Francis Group, LLC

be successful. The purpose of the release review subprocess is to allow the RM to review the prerequisite artifacts and gain knowledge of the release. During the knowledge transfer of the release strategy, the RM will work closely with the RPM to ensure a complete understanding and gain the necessary working knowledge.

Once the knowledge transfer has been completed and the RM has the necessary working knowledge of the release, the RM will prepare for the release planning meeting. Preparation for the release planning meeting will include a review of the release checklist and the determination of the deliverables and artifacts needed for the specific release. This will right size the RLC for the corresponding release; careful consideration must be given to the deliverables required to ensure that what is delivered is value added.

The RM will also review the enterprise release calendar to determine if the planned release implementation date fits within the other scheduled releases and maintenance windows. If the desired implementation date does not fit within the existing schedule, the RM will need to have discussions with the RPM to determine what can be done to meet the desired implementation date. This may entail the engagement of other RPMs of conflicting releases to gain resolution. The release implementation date will then be entered onto the enterprise release calendar. If the release is a minor release or major release of an existing service, a review needs to be done by the RM to ensure the alignment of the predetermined release schedule for that service. If it is outside of the agreed upon release schedule, justification must be reviewed to determine if an exception is needed; if it is not, adjustment of the implementation date must be made to fit the predetermined release schedule.

The RM in conjunction with the RPM will determine what resources need to attend the release planning meeting and will schedule the meeting.

Step 1.2 Planning Meeting

During the Initiation phase of the RLC it is important to ensure the needed resources have an awareness of the strategies of the release and understand what their roles are in the development, building, and delivery of the release. It is also important to solicit their input to determine their availability and the time it will take them to perform their part of the process; the planning meeting provides a forum for this type of input.

The planning meeting is facilitated by the RM and supported by the RPM. During the meeting the goals, objectives, and strategies of the release are presented, and input about the approach and how the release is going to be built and configured is received. These inputs are captured by the RPM and will be used to create the release plan. During the planning meeting, the release delivery checklist will be reviewed and the deliverables will be discussed to ensure they are right sized for the specific release. Additional topics of the

© 2010 by Taylor and Francis Group, LLC

planning meeting are the release schedule, resource utilization, and constraints that could cause a delay.

Step 1.3 Release Plan

The RPM will take the inputs from the planning meeting, incorporate them into the master project plan being created in the project management process, and will produce a release plan. The release plan consists of a release project plan, an agreed upon release checklist of deliverables for the release, a roadmap of the high-level tasks that need to be completed, a Quality Assurance (QA) approach, and an implementation strategy. The release plan can be a single document or it could be a consolidation of several documents.

The release project plan is a subset of the project plan created for the project management process; it contains only the deliverables and timelines for release deliverables. The release deliverables need to be incorporated into the master project plan. This is an area in which project management and release management complement each other. The project plan may also need to be adjusted to fit the enterprise release calendar. Alignment with the project management office and release management is necessary to make the delivery of the project sync with the release calendar.

Reusability is one of the benefits of a good release process. During the creation of the release plan, it is important to create a testing or QA approach that will provide a roadmap for testing of the release during the entire RLC. While QA is a separate subprocess, testing takes place throughout the entire RLC at some level. Being able to reuse the testing strategy throughout the RLC provides continuity of quality objectives.

The final step of the release plan is to circulate it among the stakeholders for review to ensure that all the pertinent tasks and activities had been captured. It is also important that agreement with the plan is obtained.

Step 1.4 Release Initiated

All Initiation phase activities and required documentation have now been completed, the RM is fully engaged, and tasks have been scheduled. The release plan can now formally begin to be executed. If a release management tool is being used, the RM will open a release record within the tool. If no tool is being used, the RM should create a tracking plan associated with the release project plan to track all activities. Additionally, if a document repository is available, a dedicated section should be created for the artifacts created for the release. These documents will need to be reviewed by change management and others prior to implementation of the release. The artifacts will also be used during the operation of the release and reused for the next release of the service.

© 2010 by Taylor and Francis Group, LLC

Benefits

Understanding the benefits of a process and being able to explain those benefits to others will increase the success of the process. The Initiation phase is the entrance into the release lifecycle and the beginning of creating quality operationally ready releases. During the Initiation phase the foundation of the release being developed is created. How this foundation is laid can determine the success of the release and will have long-term effects on the entire IT service. Benefits of the Initiation phase can be summarized as follows:

- Ensures project approval prior to building of environment
- Creates proper release resource planning and coordination
- Provides a process to ensure functional and business requirements have been clearly identified and documented
- Sets standards for a documented release plan guiding the building, testing, and delivery of an on-schedule release
- Creates an awareness and acceptance of the release's strategy and objectives in the resources involved in the release
- Ensures adherence to the release management policy

Artifacts and Documents

The artifacts and documents used and generated in this phase of the RLC will be a combination of work products generated specifically for the release and others generated for other processes such as the approval process and the project management process. The goal is reusability and efficiency. If a document can be used for more than one purpose, use it. Too many times, processes are not aligned in organizations and the same information is asked for many times in many different forms. This creates inefficiency and frustration on the part of the resources that have to continue to supply the same information over and over.

Document	*Who Provides*	*Description*
Release deliverables checklist (Appendix C)	RM/RPM	A checklist of artifacts needed to complete the associated release. Some entries will be prerequisites to enter the process and others will be required deliverables. The deliverables will be customized and agreed upon for each release.

© 2010 by Taylor and Francis Group, LLC

Document	*Who Provides*	*Description*
Business requirements document (BRD; organization-specific document)	PM	Provides the project requirements of the business customer. These requirements will guide many aspects of the release.
Project charter (organization-specific document)	PM	Provides an overview of the strategy, goal, and objectives of the project and associated release.
Release project plan (organization-specific document)	RPM/PM	Creates a roadmap for what release tasks and deliverables need to be completed, who will complete them, and when they need to be completed. Can be included on the master project plan or a separate project plan.

Release Configuration and Build Phase

During the RLC process there will be parallel activities taking place within a phase as well as activities from different phases taking place at the same time. Two phases that normally occur in parallel are Release Configuration and Build and the New Environment Request phases. The holistic nature of release management provides for development of both infrastructure components and software, and these two phases address these two development areas.

Release Configuration and Build is the phase of the RLC which has direct integration with the software/application development function (see Figure 5.3). As with other phases within the RLC, the role of release in the Release Configuration and Build phase is one of governance and oversight. While the actual development of the application is somewhat black box to release, the touchpoints during this phase include alignment and verification of release configuration and functionality with the strategy and business requirements of the release. It is paramount to monitor the quality and alignment of the release build with the specified functional requirements and defined quality standards of the release. It is more costly to have to rebuild an application the second time than it is to get it correct the first time. Release management's role is to monitor the

© 2010 by Taylor and Francis Group, LLC

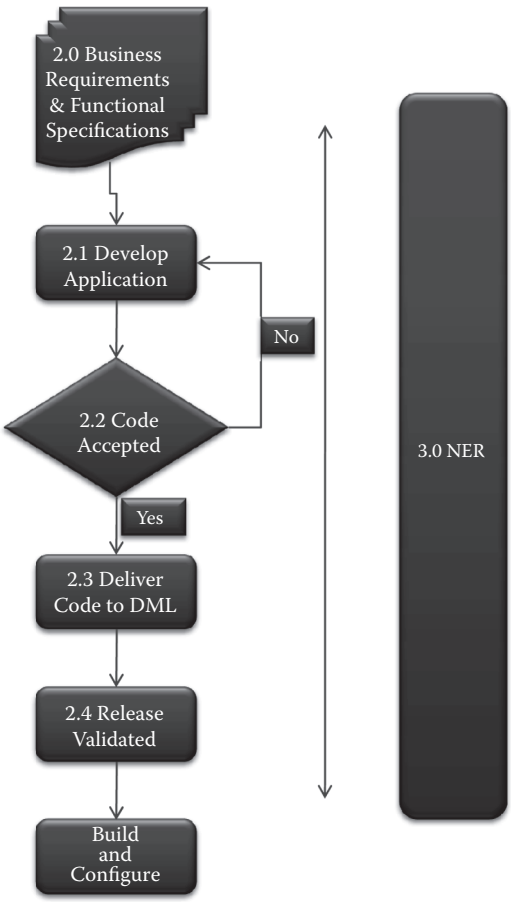


Figure 5.3 Release configuration and build phase.

build progress and quality of the build. This is done through the review of unit testing of the application.

During any process, any resource can play a role regardless of whether the role resides within the same area or another area. In the Release Configuration and Build phase, many activities will be completed by application development resources; however, they are playing a dual role—the role of application development and the role of release development. This becomes apparent during the review of the steps within this phase.

© 2010 by Taylor and Francis Group, LLC

Step 2.0 Business Requirements and Functional Specifications

During the Initiation phase of the RLC the business requirements and functional requirements for the release were captured, created, documented, and approved. Typically there is a period of time between the time the release is approved and the time actual configuration and building of the release begins. This time period can be weeks or months depending on the organization. During this period the business strategy or requirements may have changed; therefore, it is prudent to review and confirm the business requirements and functional specifications prior to beginning the Release Configuration and Build phase of the RLC. Since both were clearly and concisely documented, reviewed, and approved, the confirmation should not be a long process. This is simply a clarification and confirmation that nothing has changed.

The process of confirmation of business requirements and functional specifications is done by the RPM and the application development team. Any modifications need to be documented and recorded as a change request to the original specifications. If there is a change, the changes need to be approved by the Release Control Board (RCB) and other organizational project governance groups. The changes in requirements need to be reviewed by the RCB because the change may modify the release timelines and implementation.

Step 2.1 Develop Application

In terms of release management, the way applications are developed is black box; release management is concerned with the quality and functionality coming out of the black box and that the product meets the requirements and needs of the customer. Validation and acceptance comes in future steps.

While the actual development process of the application, or the "how," is not the concern of release management, the "who" of developing the release is. Why is this significant? It is significant in regards to the way and how extensive code review and testing are completed. If the application development is being done in-house or by a trusted business partner with which the organization has experience, one level of review and testing would be done. If the application is being developed by a new development company, then more extensive review and testing should be done. A third application model that requires a modification of review and testing is that involving an application service provider (ASP). Another emerging technology model is SaaS, or software as a service. While most would not see the implementation of SaaS as a release, it is a configuration component or service being added to the environment and the implementation should be handled as a release.

During the application development phase, release management is concerned with ensuring that testing requirements are captured and are effective in determining whether the application will perform to the desired specifications. The initial test plan, or Master Test Plan document, is created during the Initiation phase by

© 2010 by Taylor and Francis Group, LLC

the RPM and is both a project document and a release document. In the application development phase, the test requirements and test cases are validated and expanded.

Procurement of developed software or complimentary software, such as middleware used in service-oriented architecture (SOA), needs to be considered in this phase. The RPM will need to become familiar with the organization's procurement policies and processes to ensure timely delivery. Procurement timelines and activities need to be considered in the creation of the release plan.

Step 2.2 Code Accepted

After the code for each component is developed, it should be tested and verified according to the defined master test plan and acceptance criteria. The results of the test cases need to be documented; any bugs discovered need to be categorized, prioritized, and logged. The number and severity of bugs discovered during testing has a direct impact on the performance and quality of the release. Test results should be documented in the master test plan by the QA team or the resources completing the tests. Ownership of the master test plan resides with the RPM. Once the master test plan has been reviewed by the RPM it is submitted for review. Test results of all components need to be formally reviewed and accepted by the application development manager and the RM. All functionality and components need to be reviewed to determine whether they meet the required specifications.

If all criteria and specifications are not met, the code and application should be referred back to the development team for further development and correction.

Step 2.3 Deliver Code to DML

Most organizations have some type of a definitive media library (DML), a storage repository for original or gold copies of software code or programs. The repository is for internally developed code and software or purchased software; the code and software in the DML is the baseline on which new development is based. Normally, a DML is a combination of some type of virtual storage and physical storage. The variables are how organized the storage mechanism and how secure the DML is. It is not unusual for organizations not to have a DML or DML process. For the purposes of this discussion, the assumption will be made there is a DML process in place.

After the new software code or new software configuration item (CI) has been tested and verified, it is ready to be included in the build integration and unit testing. The code has been determined to be stable, and it becomes the new baseline and is deposited, along with the related documentation, in the DML. The configuration management database (CMDB) is also updated to reflect a new CI or the existing CI attributes are updated.

The DML is managed by configuration management, however the DML is "owned" by release management. In other words, release management determines

© 2010 by Taylor and Francis Group, LLC

what and when code and software is entered into the DML. The release manager reviews the test results validating the stability of the developed code. Once this is completed, the RM approves moving the code to the DML. This version now becomes the baseline that is used for development and building of the release. Even though the new code is entered into the DML and becomes the baseline, previous baseline versions are maintained, which will provide a recovery provision if needed.

Step 2.4 Release Validated

Prior to building and configuration of the release, the test results are reviewed with the project team, business system analyst, and the RCB to ensure that the functionality meets the requirements documented in the business requirements document and functional requirements. The release plan should also be verified now that the software has been developed. This needs to be verified to ensure that it aligns with the developed plan.

Validation is a critical piece of the RLC and there are many opportunities during the process for review and validation by different resources and roles. It is paramount that the customer or the business customer be kept informed during the entire process to ensure alignment and management of expectations. This subprocess is one of the first alignment opportunities.

Step 2.5 Build and Configure

When looking at the Release Configuration and Build phase of the RLC, the build and configuration process may appear to be out of place because it is the last process. However, the activities completed during this process are really the activities that bring together the software that has been developed in this phase of the RLC with the environment built and configured in the NER phase of the RLC. As with other processes in this phase of the RLC, the method and procedure used to actually build and configure the release is specific to the type of system being created and the organization-specific process and standards. Release management's role is to monitor the progress of the build and ensure the environment built in the NER phase is delivered so the application development team can configure and install the code in the new environment. This activity is performed by the RPM; the RPM keeps the RM apprised of the status of the build.

During this process, documentation of how the release is being configured and built is being completed. This documentation will be part of the release notes and will also be needed for the creation of the Service Continuity and Restoration Plan. Both of these work products will need to be reviewed by the RM and become part of the release package. The RPM should understand this requirement and should be part of the release project plan.

© 2010 by Taylor and Francis Group, LLC

It has been previously mentioned that various forms of testing will be completed throughout the RLC, and this phase is no different. Testing should be completed prior to installing the code to the environment and testing should be done after installation. A final output of this phase is the results from unit testing of the release. This testing feeds into the next phase of the RLC, called quality assurance.

Benefits

Undocumented and unmonitored application configuration results in poorly functioning applications, causing service unavailability, increased support cost, and loss of user productivity. Following a defined development process will produce the following results:

- Configuration and building of software will follow a defined, consistent process.
- Quality will be improved.
- All software masters are secured in the DML.
- Scheduling of development activities will improve.
- There will be coordination between the release configuration and build and the NER processes.
- Updates to the CMDB will be ensured.
- Detailed assembly and build instructions, scripts, and release notes are documented.

Artifacts/Documents

The activities performed during the Release Configuration and Build phase of the RLC are generally performed by the application development team, and release management monitors the activities and progress. A limited number of work products are generated specifically for release; however, the work products generated are important to the operation of the release and reusable for future releases. As with all of the subprocesses in the RLC, the RPM is responsible for the generation and approval of the artifacts and documents.

Document	*Who Provides*	*Description*
Business requirements document (BRD) (organization-specific document)	PM	Provides the project requirements of the business customer. These requirements will guide many aspects of the release.

© 2010 by Taylor and Francis Group, LLC

Document	*Who Provides*	*Description*
Functional requirements document (organization-specific document)	PM	Takes the business requirements and creates technology requirements needed to deliver the stated business requirements.
Master test plan (Appendix D)	RPM/PM	Describes the testing strategy for the release and provides an artifact to record test results and approvals.
Release notes (technology- and organization-specific document)	RPM/PM	Documented release functionality and build notes are part of the release notes. These are documented in a technology- and organization-specific manner. They are critical to the reusability of release management.

New Environment Request (NER) Phase

In Chapter 4 the New Environment Request (NER) process and activities were discussed in detail; therefore, the subprocess steps will not be discussed here. Please refer to the previous chapter. However, it is appropriate in this chapter to discuss how the NER process aligns and fits in the RLC. The NER (see Figure 5.4) is a process that runs in parallel with the build and configuration process. The NER process is focused on designing and building the infrastructure environment, and the build and configure process is focused on developing the software and code. Each of the processes deals directly with the planning, building, configuring, testing, and implementing of their respective focus areas. The final phase of the build and configure process deals with joining the work products of both processes together and delivers a functioning product or application.

Although a functioning product has been created at this point in the RLC, it is not production ready; additional testing and preparation for implementation needs to be completed.

© 2010 by Taylor and Francis Group, LLC

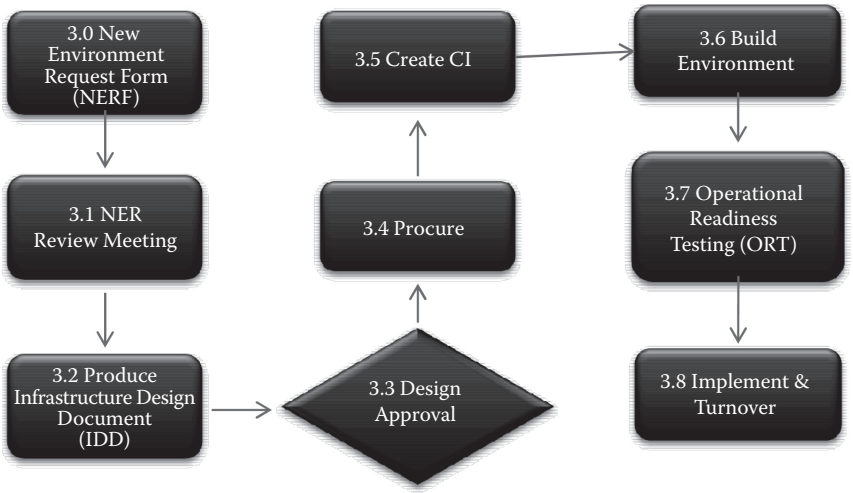


Figure 5.4 New Environment Request phase.

Benefits

Similar to the release configuration and build phase, a consistent, well-defined, and monitored process increases the quality and availability of the service. Lifting the veil and partnering with the infrastructure groups will improve the form and fit of the environment being built, and focusing on one strategy and objective will provide for less development time. The benefits of including the NER process in the RLC are:

- Clearly defined and documented requirements
- Architecturally sound environment design
- Environments are built according to predetermined specifications and time frames
- Requests for technical resources are scheduled and prioritized
- Improved coordination and communication between project and technical groups
- Increased quality, which translates to increased availability and reliability

Artifacts/Documents

The NER process has artifacts and documents as work products, as do the other processes. Release management plays more of a facilitation role in the NER process than in the configure and build process, therefore the RPM and RM are more involved in the completion and review of the documentation. Even though the RPM and RM have more involvement, the completion of the documents is done

© 2010 by Taylor and Francis Group, LLC

by the technical and project resources. The RPM has a strong facilitation role and ensures that the correct resources are engaged. The RM plays the role of facilitator by ensuring that artifacts and documents are completed and facilitating review meetings.

Once again, the work products created in the NER process are important to the operation of the release and reusable for future releases.

Document	*Who Provides*	*Description*
New environment request form (NERF; see Appendix E)	RPM/Tech Lead	Artifact used to enter the NER process. Provides release overview and architectural review.
Infrastructure design document (IDD; see Appendix F)	Tech Lead/RPM	Used to provide a more in-depth overview of the release environment that needs to be built. Is used to gain approval to procure hardware. Supported by the functional specifications document.
Security risk assessment (see Appendix G)	RPM	Used to obtain a security risk assessment of the release to identify risk and mitigation strategies.
Functional specifications (see Appendix H)	Infrastructure Iteams/Tech Lead	Contains detailed functional specifications of the release, and is used by the contributing infrastructure teams. Used to supplement the IDD. Also used to provide guidance for the environment build.
Operational readiness testing (ORT) manual (see Appendix I)	Infrastructure Iteams	Used by the infrastructure operations teams to ensure the environment being implemented has been configured correctly prior to it being put into production. The ORT manual is used twice—during the NER process and during the Operational Readiness (OR) process. The ORT manual includes the key application guide, which provides information to the operations team.

© 2010 by Taylor and Francis Group, LLC

Quality Assurance Phase

A quality release is one of the core goals of a defined, well-executed release process; the RLC is a structured approach and serves as a catalyst to focus on quality delivery. Quality assurance needs to be embedded throughout the RLC; it's not a sequential step in the RLC, even though it is represented as the fourth phase. The Quality Assurance (QA) phase (see Figure 5.5) runs parallel to the other processes. A well-defined QA process will provide a production-ready release.

During the Release Configuration and Build and NER phases, unit testing and system testing takes place. The results of these tests should be documented in the Master Test Plan and will be reviewed prior to implementation. The Master Test Plan is a document used throughout the RLC and should begin during the initiation process. The Master Test Plan should include the various types of testing done during the release. It should also include the test cases to be performed, and these test cases should map directly back to the business requirements. The Master Test Plan will also include performance testing as well as user acceptance testing (UAT). The testing process may identify bugs

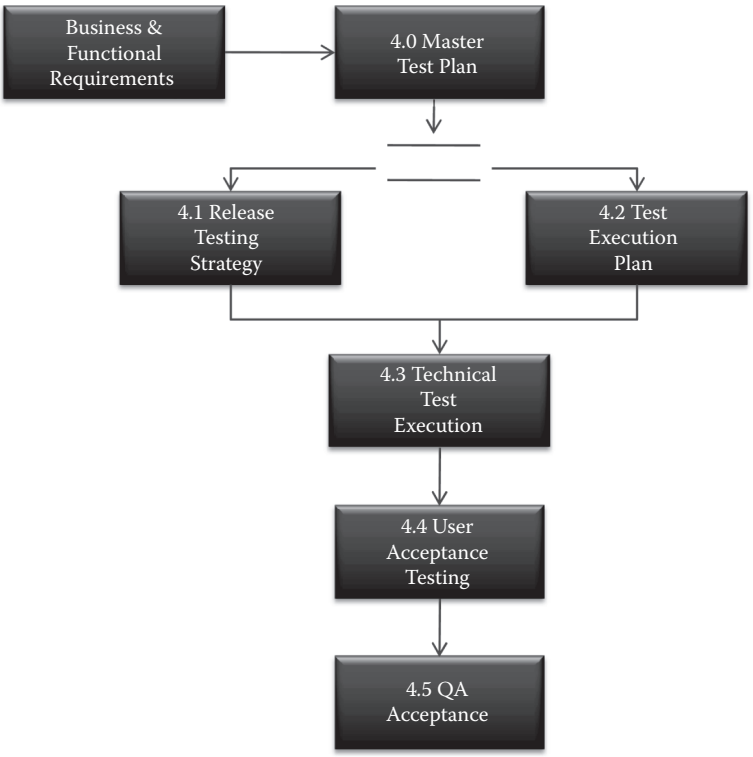


Figure 5.5 Quality Assurance phase.

© 2010 by Taylor and Francis Group, LLC

within the release; the project team must review these bugs and determine fixes or workarounds.

The RLC requires that the Master Test Plan be reviewed by the quality assurance group (if this role exists in the organization) to either endorse the release or reject its implementation. Part of the acceptance review is to ensure that any potential bugs that may cause issues in production have been properly identified and mitigated. The report also ensures that the project team has acknowledged the associated risk.

Step 4.0 Master Test Plan

When a professional soccer team steps on the pitch at the beginning of a match, the team has a game plan on how they are going to execute their strategy and plays; the plan anticipates risks and provides a plan for mitigating those risks. During the match, the coach reviews the plan and documents the successes and failures of the team. At halftime, the coach will review the plan to determine if any new risks have arisen, and if so, how the team can mitigate the risk or if it can be accepted.

Much like the coach using a game plan to play, test, and manage the risk of the soccer match, the QA process performs a similar function. The game plan for the QA process is the Master Test Plan. The creation of the Master Test Plan begins during the Initiation phase of the RLC. Inputs are received concerning the business requirements and functional requirements and serve as the basis for the Master Test Plan. Understanding why the release is being created and what business problem is being solved is critical to creating a sound testing strategy and plan. Accurately recording this information is paramount and sets the stage for the creation of a successful test plan and strategy.

The RPM will facilitate the creation of the Master Test Plan and will gather the appropriate resources to ensure effective and complete testing. If the organization has a quality assurance group or practice, they will also be engaged. Depending on their role in the organization, the QA group may actually perform some or all of the testing, or the QA group may be an oversight and governance body.

Step 4.1 Release Testing Strategy

The next two processes in the QA phase deal with continued creation of the Master Test Plan—the testing strategy of the plan and how the test plan will be executed. Both of these concepts are critical to the success of the release.

The testing strategy must be based on the business strategy and ensure the goals and objectives of the release are achieved. To ensure a successful outcome of the release, a planned and structured approach must be taken. Creating the release testing strategy begins with understanding the critical success factors needed to meet the desired results. Examples of critical success factors may be system performance, transaction processing time, or ease of use of the service. The critical success

© 2010 by Taylor and Francis Group, LLC

factors need to have a direct correlation to the goals and objectives outlined by the business. The critical success factors should not be limited to only those that have a direct correlation to the business; there are critical success factors aligned to IT standards and specifications. These IT-critical success factors include failover capabilities, server capacity utilization, and network performance standards.

Once critical success factors have been established, the method of testing for compliance needs to be created. The typical method of testing is the utilization of test cases. Developing the structure of test cases is dependent on the type of testing and the desired outcome. As an example, if performance testing is being done, the desired outcome may be the need for 400 concurrent users to be logged on at the same time with 30% server utilization. The test case could be written to have user sessions created, 25 sessions at a time, every 15 minutes, until the desired number of users is realized. At the same time, server utilization will be monitored to determine if the goal of 30% is exceeded. If it is exceeded, then the test fails to meet the desired outcome.

A second facet of creating a testing strategy is when and how the testing will be executed. Will the testing be done during normal business hours or will it need to be done during a maintenance window or after business hours? If system stress testing is being done, utilization of the production environment may be necessary. If this is the case, a strategic decision would be to complete the testing after hours so the additional network traffic created by stress testing would not have an adverse effect on users.

Determining how the testing will be done is also a strategic decision that needs to be made when creating a testing strategy. In some types of testing, a testing tool may be used to simulate system processing. When planning for user acceptance testing, the users who will be doing the testing need to be scheduled and trained on the system or application. These are the strategic decisions that go into creating the release testing strategy.

Step 4.2 Test Execution Plan

Creating a release testing strategy focuses on providing needed critical success factors to ensure a quality release is created and the established goals and objectives are met. A second part of creating an effective test plan is to determine how the created tests will be executed. In the previous subprocess, some of the execution considerations were discussed. Considerations such as how the test will be completed, who will execute the test, and when the test should or can be executed are as important as the testing strategy.

The how, who, and when can have different answers depending on the type of testing that is being done. A primary consideration is what impact the testing will have on the production environment. In the earlier stages of development, configuration and activities are normally done on subproduction environments; therefore, there is normally little or no impact on the production environment that would have an adverse impact on users. However, as development continues and testing

© 2010 by Taylor and Francis Group, LLC

needs to utilize the production environment in some form, increased consideration needs to be given to the impact on users.

How the testing will be completed also needs to be planned and documented in the Master Test Plan. Some types of testing will require utilization of test tools and testing environments. Organizations typically do not have unlimited testing tool capacity and environments; therefore, there is a need for scheduling as to when the tools and environments can be used for testing.

Besides planning how the testing will be completed, the question of who is going to do the testing also needs to be planned. When planning for user acceptance testing, consideration needs to be given to how much testing needs to actually be done by the end user. Can a business systems analyst do an adequate job of testing or do end users need to be involved? It is always a good practice to have at least some end user testing, simply because how the end user utilizes the system may be completely different than the way in which the development team perceives that it will be utilized. This is a primary reason why the user acceptance testing (UAT) test cases need to be reviewed by a representative sampling of the end user community.

A final consideration in creating the Master Test Plan is how the test results will be documented. Again, how and what type of testing will be done will have an influence on how the results are documented. Testing done using a testing tool will have a reporting mechanism in place. However, UAT will need to have a method to ensure results are consistently documented by all participants. A traceability matrix is one method that can be utilized. A traceability matrix is used to trace the results of test cases to the critical success factor. Without the ability to determine clear traceability, determining if the critical success factor is met will just be speculation.

Release management's role in creating the release testing strategy and the execution plan is one of facilitation and oversight. The responsibility of the Master Test Plan and facilitating the creation of the testing strategy and execution plan falls to the RPM. The RPM may not actually write the plan; however, the RPM will ensure that the resources needed to create a comprehensive plan are engaged. The RPM reviews the plan for completeness and coordinates the review and acceptance with the RM.

Step 4.3 Technical Test Execution

Earlier in the discussion of the Quality Assurance phase, references were made to different types of testing done during the release and development lifecycle. The types of testing can be separated into two types, technical testing and user testing. Technical testing deals with ensuring the infrastructure and applications function from a technical perspective. Are the infrastructure components operating as designed? Can server performance be maintained when the system is being used by the required number of users? Does server load balancing function as designed? These are just a few of the questions to be answered. User testing deals with usability and functionality of the services from the end users' perspective.

© 2010 by Taylor and Francis Group, LLC

Responsibility for completing technical testing rests with either the quality assurance group, if one exists within the organization, or with the technical group completing the development. It is recommended that a separate group from the development team execute the testing as designed in the Master Test Plan. This provides for an objective view and opinion of the test results. This may not be possible, however effort should be made to separate as much of the testing as possible. Release management's role in technical test execution is one of awareness and review. The RPM monitors the execution of the different types of testing from the perspective of ensuring the release schedule remains on track. The RPM also monitors the test results to ensure the testing objectives are met; special attention is given to the number of defects being generated from the testing. If an abnormal or unacceptable number of defects are being generated, the RPM needs to raise this as an issue to the RM. The RM can take the necessary actions to mitigate the risks being raised by the number of defects by discussing the issue with the Release Control Board (RCB). The RM will need to review the final test results prior to implementation of the release.

There are various types of technical testing done during the development of a release, and the types of testing depends on the type of technology being developed and the testing ability and standards of the organization. Some general types of testing include smoke testing, regression testing, performance testing, and failover testing. Since this is a book about release management and not technical testing, there will not be a detailed discussion of how to complete these types of testing.

Step 4.4 User Acceptance Testing

Prior to release acceptance and implementation, testing must be completed to ensure that the release meets the desired requirements and functionality of the user. There have been times when a release has moved through the development cycle and is being readied for implementation without the users testing the functionality. When users finally get an opportunity to test the functionality, the release does not function as required. There can be several reasons for this, however a key cause is that there was no release process in place and a sound testing plan was not executed. The results of this situation are increased costs associated with reworking the release and increased time to delivery.

User acceptance testing is essential to ensuring the release meets the goals and objectives of the customer and the end users. UAT should be done throughout the development process, not just prior to implementation, as the example above demonstrated. Traditionally, UAT is thought of as an activity to gain acceptance of a release's functionality prior to implementation. Although this is one type of UAT, there are other types. The other types take place in the form of reviews of beta functionality, review of design documentation, or a review of a proof of concept. To get the full value of these types of testing and reviews, objectives need to be established. Another consideration is defining who is a "user." A user can be the

© 2010 by Taylor and Francis Group, LLC

end user or someone familiar enough with how the release being developed will be used; in some instances, a business systems analyst can fill this role.

Release management's role in UAT is similar to the role it plays in the technical testing. One significant difference is that users testing the functionality of the release may need training on how to use the application or service. This is especially true if the release being implemented is new to the organization. This may also be the case if the release is introducing new functionality to an existing service. Without this training, the user may not have any reference or concept of how the application is designed to function. From the RPM's perspective, this introduces additional tasks and documentation to manage. The training of resources bleeds into an activity in the operational readiness process. Training documentation can be reused when training other resources.

Results of UAT are a significant milestone for release management as it demonstrates the customer's acceptance of the release. The results need to be documented and are used in completing the traceability matrix to demonstrate that the release has met the business requirements and the defined goals and objectives of the release. Once this has been demonstrated, the RPM will ensure the customer's review and approval is obtained.

Step 4.5 QA Acceptance

The final step of the Quality Assurance phase is the completion and acceptance of the Master Test Plan. The RPM will ensure that all of the types of testing that are documented within the plan have been completed and the approved critical success factors have been met. Risk mitigation is a concern of release management, and the Master Test Plan should document any unresolved defects discovered during the various types of testing. It is generally recognized within IT development that there will be some level of defects. These defects must be analyzed and categorized into the level of risk the defects introduce into the service being developed. These defects and the level of risk must be reviewed with the customer for acceptance or nonacceptance. The accepted defects become known errors and are documented in the Master Test Plan. The known errors will also become part of the support and escalation documentation used during the operation of the service. The known errors should also be documented in the known error database (KEDB). Release management's role is to ensure that these defects have been documented and reviewed with the customer. This must be done prior to implementation.

Once the Master Test Plan has been reviewed and approved by the QA group, the RPM, the customer, and the RM, the RM will submit it to the Release Control Board (RCB) for review and approval. Once approved, it becomes part of the permanent release record and will be stored within the release management database or tool for reference and reusability. Change management will need to review the Master Test Plan prior to submitting the request for change (RFC) for approval to the Change Advisory Board (CAB).

© 2010 by Taylor and Francis Group, LLC

Benefits

The core benefit of the Quality Assurance phase is that the quality of the release being developed and implemented is assured. This is done through creating critical success factors that have a direct correlation to the business requirements of the release. Being able to create a clear and concise test plan that demonstrates the critical success factors have been met is a key benefit. Other benefits of the Quality Assurance phase are:

- A documented test plan with clear and concise critical success factors is created.
- All releases are thoroughly tested according to business and functional requirements.
- Alignment between functional specifications and business requirements is ensured.
- Identification and documentation of known errors is ensured prior to release into production.
- A mechanism is created to demonstrate customer and user acceptance of the release.

Artifacts/Documents

The Quality Assurance process is a critical part of the RLC because it is core to ensuring the quality standards of the release being developed are maintained. A critical part of ensuring that the quality standards are maintained is being able to establish a strategy and document that strategy. Being able to document a testing roadmap is critical to ensuring that quality standards are maintained. The documentation produced during the QA process, such as a Master Test Plan and traceability matrix, are critical tools used by the RPM and the RM to ensure the established critical success factors are met. There are many other artifacts used by QA groups, but these are out of the scope of release management and should be used as good practices of a good quality assurance program.

Document	*Who Provides*	*Description*
Master Test Plan (see Appendix D)	RPM/PM	Describes the testing strategy for the release and provides an artifact to record test results and approvals.
Test Strategy (organization-specific document)	RPM	Documents the testing strategy and its execution. It can also define critical success criteria of the types of testing.

© 2010 by Taylor and Francis Group, LLC

Operational Readiness Phase

Organizations typically do an adequate job in creating and building new applications and systems, however they fall down in providing knowledge to support the release once it is implemented. Necessary documentation must be provided to support the new IT service in operation once it is implemented. This is a common issue if there is not a good release practice in place. This is an area of discrepancy between the objectives of release management and project management, and was discussed in Chapter 3. Release management is focused on the lifecycle of the entire service and project management is focused on delivering the requirements defined for the specific project. The Operational Readiness (OR) phase (see Figure 5.6) is primarily focused on preparing the release for operational turnover and ensuring the necessary artifacts and resources are in place to support the service once implemented.

Ensuring the release is operationally ready is a core responsibility of the release process. As with other phases in the RLC, the creation of the artifacts is facilitated by the RPM and reviewed by the RM, however during the OR phase, the RM is clearly focused on whether the release can be supported during operation. In other phases, the RM's review of work products is designed to ensure completeness and that approvals have been obtained from subject matter experts. In the OR phase, the RM must ensure the viability and usability of the work products. This may create spirited discussions between the project team and the RM, primarily due to the differences in scope.

Step 5.0 Operational Readiness Entrance

Entrance to the OR phase is really the entrance to the beginning of the implementation and turnover of the release from development to operations. Determining if the release has reached the stage where it can be prepared for implementation is determined by whether all of the task and milestones have been met in both the release plan and the project plan maintained by the RPM. Determination of preparedness is demonstrated through the sign-off and approval of the Master Test Plan by the business and the other stakeholders in the release. The Master Test Plan should contain evidence that the requirements of the release have been tested and that the critical success factors defined at the beginning of the release have been met. The release should not move forward until the approval of the customer has been received. Moving forward without this approval increases the risk of dissatisfaction with the acceptance of the final product and could cause additional financial ramifications.

The next steps in the process focus on various facets needed to ensure efficient support and operation of the release. It is not unusual for project teams to neglect the preparation of the work products until the end of the development cycle, however these concepts should be reviewed during the planning stages of the release and should be developed during the RLC.

© 2010 by Taylor and Francis Group, LLC

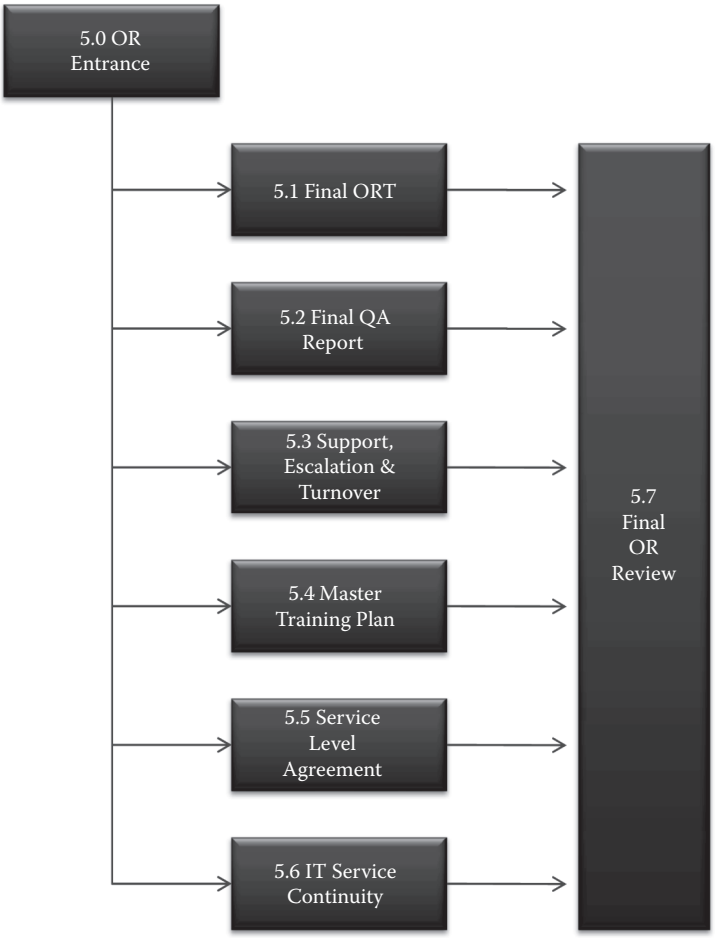


Figure 5.6 Operational Readiness phase.

Step 5.1 Final Operational Readiness Testing (ORT)

At first glance, the ORT step may seem repetitive of the ORT step in the NER phase and it may seem unnecessary. However, there are portions of the phase that need to be performed for the second time. There are also portions of the ORT that will be new. Much like the first ORT, the purpose is to ensure the release is technically sound and ready for operation in production. The actual ORT work will be done by the technical infrastructure team; release management's role during ORT is to facilitate and coordinate the execution. The actual activities to be completed during the ORT are organization specific due to the specification and requirements. From a generic perspective, items to be considered during the ORT are ensuring virus detection software is installed, detection and monitoring software or agents

© 2010 by Taylor and Francis Group, LLC

are installed, ensuring all operating systems are up to date with the most current patches, and documenting all configuration items and relationships in the CMDB.

Facilitation and coordination of the ORT by release management includes the review and approval of the ORT Manual, ensuring a request for change (RFC) has been raised to change management for approval of the activity, and coordinating with the technical team to ensure resources can be scheduled. It may seem that release management is assuming a role of oversight in the process, but it is really a role of facilitation and acting as a central point of contact. Once the ORT is completed, a sign-off from the technical team is needed indicating completion. This approval, captured in the ORT Manual, becomes part of the release document archive. It will be referenced in future releases of the service.

Step 5.2 Final QA Report

The Master Test Plan has been referred to several times during the various processes in the RLC. This document serves an important role in documenting the strategy, the critical success factors of the release, the various testing and results, and the final acceptance of the release based on user acceptance testing. The Master Test Plan must be created early in the lifecycle and must be used extensively throughout. The RPM must be diligent in ensuring that it is updated as the stages of testing are completed.

Review of the Master Test Plan is a prerequisite for entry into the OR phase, but the final review of the Master Test Plan is completed after the ORT is completed. This ensures that if any type of regression testing or any other type of testing is done during ORT due to issues, the results are captured and reviewed. Once reviewed, the Master Test Plan becomes a part of the release archive and will be referenced by change management during the final approval prior to implementation.

Step 5.3 Support, Escalation, and Turnover

Unless the organization is mature and is using some type of release process, it is typical for the discussion of how the service will be supported in operation to be left until the end of the development cycle. When this happens, the work product produced lacks quality and is usually not comprehensive enough to provide the support organization enough information to provide efficient support. Another aspect is the planning of a concept called *early life support*.

Early Life Support

No matter how good the preparation for the implementation of the release is, there will always be a need for some type of focused support during the initial days of the release. *Early life support* is a concept that deals with the initial days or weeks following the implementation of the service and how to best provide the support. The length of this period can vary depending on whether the release is a new service or a minor release

© 2010 by Taylor and Francis Group, LLC

and the complexity of the release. Planning of early life support can be a complex activity as it involves the coordination of resources, facilities, and procedures. Early life support can determine how successful the implementation of the release will be.

Figure 5.7 illustrates a high-level view of an early life support timeline. The time frame illustrated is an example, and the times can be increased or reduced depending on the needs of the organization.

Pre-Implementation Activities. Consideration of pre-implementation activities is based on the initial support approach for the release. The approach may be to turn the entire initial support over to the service desk or provide a centralized support mechanism; the example illustrated is a model for the latter. In this model there will be many activities that need to be planned for, and the planning needs to begin several weeks prior to implementation as lead times may be required. A documented strategy should be created for early life support and turnover activities. The strategy should include:

- Will a "war room" be used, and if so, how long will it be active?
- What is the roll of the service desk during the transition period?
- How will the war room be staffed?
- What types of communications facilities are needed?
- How will issues be logged, addressed, and communicated to the service desk for long-term support?
- How will bugs that are discovered be addressed?

These suggested activities are thought-starters; additional details will arise as these are considered.

Post-Implementation Activities. Activities executed during post-implementation are planned for within the early life support strategy. It is important to maintain the strategy set forth and not stray due to lack of interest; it is paramount to the success of the implementation that the resources committed remain committed. It is also important to be ready to extend the length of early life support if it is found that it is needed.

The final activity of early life cycle is the turnover of support to the service desk. This activity should include a final review of support scripts to ensure any learning captured during early life support is recorded for future reference. The RPM and the service desk manager should complete a final turnover assessment to complete the turnover.

The RPM is responsible to ensure that the early life support strategy and support is considered and in place prior to recommending to the RM that the release is ready for implementation. If the RM receives a recommendation for release implementation that does not have a provision for early life support, it should be referred back to the RPM for completion.

Support and Escalation. Understanding how the service and releases to that service will be supported in operation is a primary concern of release management. It was previously mentioned that project management many times can become solely

© 2010 by Taylor and Francis Group, LLC

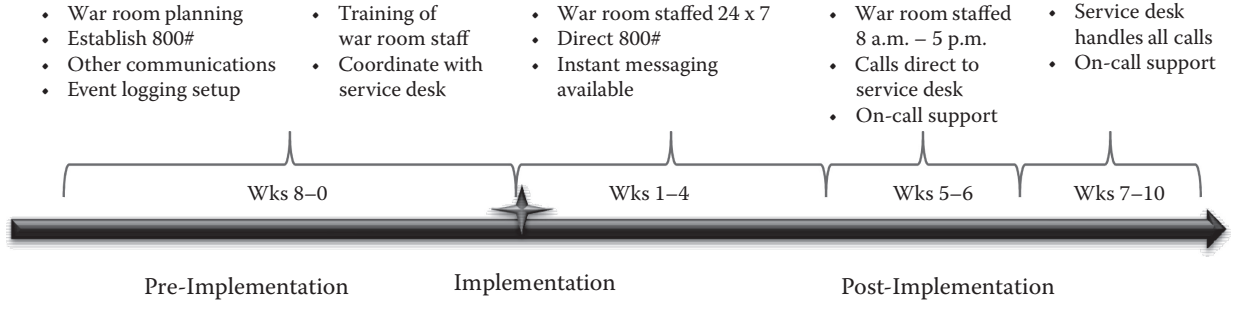


Figure 5.7 Early life support.

© 2010 by Taylor and Francis Group, LLC

focused on delivering the project and will not consider how to support the release until it is time to implement. While this is understandable, it increases the responsibility of the RPM and the RM to ensure support scripts and escalation procedures are in place prior to implementation. While the RPM does not have the responsibility to create the support scripts and escalation procedures, the RPM and RM do have the responsibility to review them to determine coherency and usability.

In addition to reviewing the scripts and escalation procedures, the RM must ensure the service desk has reviewed and accepted the work products. Many times when the created documents are reviewed by the service desk and level 2 and 3 support groups, there will be questions that need to be answered. The documents may need to be modified and rereviewed. Sign-off of acceptance is a requirement, and the RM will secure approval prior to recommending release implementation.

Step 5.4 Master Training Plan

Development teams often overlook the training aspect of implementing a new release. If training is considered, it is usually focused at the users of the release, and oftentimes the service desk and support resources are overlooked and left to try to support the release using the support documentation. Understanding how the application or service functions improves the ability to efficiently support the application.

The Master Training Plan provides a strategy for how and who should be trained on the release being implemented. The strategy and approach will be documented in an artifact named the Master Training Plan.

The strategy should not only include who should be trained but also a mechanism for the training. Will the training be done via the Web or in a classroom? Each will require various levels of preparation such as training curriculum and media. Will different curriculum need to be developed for different audiences? What is the right timing for the delivery of the training? Once again, the resources involved need enough lead time to properly prepare and deliver the training strategy. The role of the RPM is to ensure that the training strategy and training materials are developed, and that training is scheduled and delivered to ensure resources are adequately prepared for the release implementation. The RM will review the Master Training Plan to ensure the training strategy has been completed prior to recommending implementation of the release.

Step 5.5 Service Level Agreement

Key to managing a service is being able to set expectations, gain agreement, measure performance, and use the data for improvements. The way this is done is by creating service level agreements. There is certainly room for discussion on whether to call this agreement a service level agreement (SLA) or an operational level agreement (OLA). The difference can be discussed within the specific organization, however the term *SLA* will be used in this book.

© 2010 by Taylor and Francis Group, LLC

Service level agreements are used to set expectations with the service's customer with respect to the expected level of service delivery. Within a SLA, measures should be set for hours the service will be available, the expected percentage of compliance with the hours of availability, the restoration objective when the service experiences outages, and any scheduled maintenance periods when the service will be unavailable. Setting of these expectations begins during the design phase of the release when the service level requirements are captured in the business requirements. The requirements captured will provide guidance for system specifications that will lead to building the release to the needs of the customer. These requirements then translate into the service levels. The service levels will be documented and agreed upon by both the customer and the groups delivering the service.

Organizations oftentimes struggle with the creation of SLAs and many times they are not created. This is mainly because service level requirements have not been accurately captured and agreed upon. This leaves customers not knowing what to expect, which in turns creates customer dissatisfaction.

Creation of service level agreements is the responsibility of the development team and is facilitated by the RPM. If the organization has service-level management (SLM), then SLM should be engaged to assist in the creation and negotiating of the SLA. The RPM will ensure the SLAs are accurately documented and agreed upon. The RM will review the SLA to ensure the same and will forward to service level management to include in the service catalog.

Step 5.6 IT Service Continuity

A disaster occurs at the data center and there is a need to restore all of the applications running on the servers at the data center. Are there disaster recovery and restoration plans for the applications and services running in the data center? There are success stories of organizations that have been prepared and others about organizations that were not prepared. The difference is returning to operations and doing business or not doing business. Preparing for a disaster is usually the first area that is cut when budgets come into question and timelines are not being met, however this concept can cost organizations millions in lost revenue, and in some cases causes the organization to cease operations.

During the development of the initial release of a new service, it is imperative to develop an IT service continuity and restoration plan to be prepared for the unseen disaster. The plan needs to be prepared by the release development team and must include detailed information on how to restore the application or service. The plan should also include:

- Relationships to supporting and dependent IT components
- Specific server names
- Resources needed to restore service
- Established recovery-point objectives and return-to-operation objectives

© 2010 by Taylor and Francis Group, LLC

The plan must also align the IT service with the business continuity plan that the service supports. These plans must be reviewed regularly and kept up to date.

The RPM is responsible to ensure that the IT service continuity and restoration plan is created, reviewed by the dependent IT groups, and approved by the IT service continuity manager. The RM will review the plan and ensure it is accepted prior to recommending release implementation.

The initial creation of the plan will be done during the development of the initial release. During subsequent releases, the IT service continuity plan must be reviewed and updated if any changes are made during that release. The RM will require this review and update to be completed.

Step 5.7 Final Operational Readiness Review

A final step in the Operational Readiness phase is the final OR review. This may seem repetitive of the preceding reviews done during the subprocess, however a simple reconciliation of the required artifacts is needed. At the being of this process description it was mentioned that release management is heavily involved in operational readiness, which is due to the core responsibility of release management—ensuring a quality release is operationally ready prior for implementation. During the review of the various work products and artifacts for operational readiness, there will be several items that will need to be revised and reworked. To ensure these requirements are met, the final review is needed. A simple checklist is used by the RM to verify the requirements have been met.

The final task of the Operational Readiness phase is ensuring the release artifacts and documents are stored in the corresponding release repository. This repository should be able to be accessed by others so the documents being stored can be utilized.

Benefits

The focus of the Operational Readiness phase is to ensure the release is ready to release into operation, and once implemented, can be supported to optimal performance and function as designed. A central benefit of operational readiness is customer satisfaction. Other benefits derived from operational readiness are:

- Testing of the release before implementation into the production environment is ensured.
- Early life support of the release is planned and ready for execution.
- Service desk and support groups are trained and ready to support the release.
- Users and support resources are trained.
- Expectations are reviewed and documented in service level agreements.
- The organization is prepared to restore the service in the case of a disaster situation.
- Operational artifacts and documentation are stored in a central repository.

© 2010 by Taylor and Francis Group, LLC

Artifacts/Documents

During the Operational Readiness phase there are many artifacts and documents used to convey and capture information critical to the readiness and future operation of the release. These work products will be used and referenced routinely by many different resources, so it is important that they are accurately completed and understandable for the resources using them. Review by the RM will ensure their usability.

Document	*Who Provides*	*Description*
Master Test Plan (see Appendix D)	RPM/PM	Describes the testing strategy for the release and provides an artifact to record test results and approvals of the business.
Operational Readiness Testing Manual (ORT Manual; see Appendix I)	Infrastructure teams/RPM	Used by the infrastructure operations teams to ensure the environment being implemented has been configured correctly prior to being put into production. The ORT Manual is used twice—during the NER phase and during the OR phase.
Support and Escalation/ Service Desk Turnover Document (see Appendix J)	RPM	Documents support scripts and procedures to be used to support the operation of the service. Outlines early life support process.
Master Training Plan (see Appendix K)	RPM	Documents the training strategy and training plan for the release. Creates a training outline; describes who needs to receive training and how it will be delivered.
Service Level Agreement (see Appendix L)	RPM	Provides simplistic service level objectives. Provides a way to set performance objectives with the customer.
IT Service Continuity & Recovery Plan (see Appendix M)	RPM	Describes recovery procedures for the service. It provides recovery teams with an overview of the services-related components and necessary contact information.

© 2010 by Taylor and Francis Group, LLC

Document	*Who Provides*	*Description*
Release Notes (technology- and organization-specific document)	RPM/PM	Documented release functionality and build notes are part of the release notes. These are documented in a technology- and organization-specific manner. They are a critical part of reusability of release management.

Implementation Phase

The release has been built, it has been tested and accepted by the customer, it has now reached implementation. Release management's involvement in the Implementation phase (see Figure 5.8) is one of review and observation. Release works with change management to ensure the required steps for a successful implementation are completed prior to receiving approval to implement. These steps include implementation and back-out planning and scheduling, validation, and documented success criteria. These high-level steps are captured in the implementation section of the release plan documentation and recorded in the request for change (RFC).

The change management process that occurs during the implementation process will not be reviewed here; for the purposes of this book this process is out of scope.

Step 6.0 Implementation and Back-out Plan

The implementation approach should be part of the release plan along with the stated risks associated with the approach being used. The detailed implementation and back-out plan will be completed by the development team and included in the RFC that is submitted to change management for approval. The implementation plan must contain incremental success criteria that must be met as the release is being implemented. If at any time the success criteria is not being met, the implementation must be stopped and backed out, and the system must be restored to the baseline state.

Increasing implementation success of the release is a primary focus of good planning. To increase the probability for success, the implementation and back-out plan should be rehearsed in a subproduction environment. The implementation rehearsal should be done in an environment that closely emulates the production environment. The benefit of doing an implementation rehearsal is to identify any issues that may occur and resolve them prior to the live implementation.

The role of the RPM is to monitor the development team to ensure the implementation and back-out plan are comprehensive and complete prior to raising the RFC. The RPM should also raise any risk identified during the implementation rehearsal with the release and change managers.

© 2010 by Taylor and Francis Group, LLC

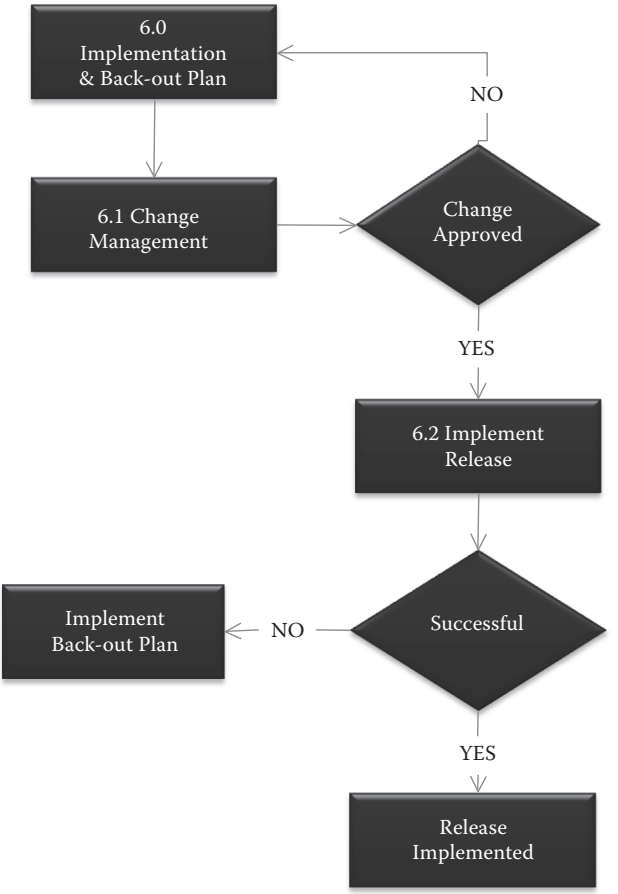


Figure 5.8 Implementation phase.

Step 6.1 Change Management

As previously stated, the actual change management process will not be discussed in this book. There is a step in the Implementation phase to illustrate the interface and relationship between release management and change management. Release management plays the role of governance facilitation. In the change management process, there is a need to ensure the release being implemented has been approved and tested, and release management will provide this verification for change management. Release management also assists with the schedule of the release implementation. During the planning phase of the release, the implementation was aligned with other release implementations and documented on the release schedule. If there has been any variance or delay, the release schedule has been adjusted. The RM will provide the information

© 2010 by Taylor and Francis Group, LLC

needed by change management to process and prepare the release for implementation approval.

Change management submits the implementation RFC to the appropriate approval authority. If the RFC is approved, the release will move to implementation. If the RFC is not approved, it will be closed and referred back to the project team for additional information. The RFC for implementation can be resubmitted.

Step 6.2 Implement Release

Once the release is approved for implementation, the RPM will reverify that the resources needed for implementation are still available. The method of implementation will be technology specific. Additionally, some organizations may have automated solutions that will schedule and implement code into the environment. The implementation methodology is outside the scope of release management.

When the implementation begins, the RPM will track the implementation task being performed. In some organizations, the RPM may be responsible for sending communications for task initiation, task verification, and completion. The RPM will also monitor the implementation milestones to ensure the goals are being met. If the milestones are not being met, the project team must make a decision as to whether the release needs to be rolled back. If this occurs, the RPM will monitor the back-out plan to ensure all of the steps are completed and the environment is returned to the baseline state.

At the end of the release implementation, whether it is successful or failed, the RPM will notify the RM of the outcome.

Benefits

Being able to strategically plan and prepare for the release implementation will significantly improve the success of the implementation, and provides the opportunity to create a back-out plan as a contingency if the implementation is not successful. The phase also supports the change management process.

- Provides for a documented, detailed implementation plan and identifies needed resources.
- Creates intermediate success factors that need to be achieved during the implementation. If the success factors are not met, the back-out plan can be executed.
- Supports the change management approval process and provides evidence of testing.

Artifacts/Documents

The artifacts created during the Implementation phase are facilitated by the RPM, but are generally created by the release project team and tech lead. This is due to

© 2010 by Taylor and Francis Group, LLC

the technical nature of the artifacts. While the implementation artifacts are release specific, there is some reusability for future releases.

Document	*Who Provides*	*Description*
Request for Change (RFC; organization-specific document)	Tech Lead/RPM	The RFC is a change management document to register and provide implementation details of the release. The RFC is the entry point into change management and requests approval for implementation of the release.
Implementation and Back-out Plan (organization-specific document)	Tech Lead/RPM	Provides a detailed plan for release implementation. Includes a list of tasks that need to be completed and lists the resources needed to complete the task. The plan also creates specific goals that need to be achieved during the release. The back-out plan is a detailed plan to revoke the implementation activities and return the service to the baseline state.

Post-Implementation Phase

The final phase in the RLC focuses on completing post-implementation activities. While some organizations consider the release completed after it is implemented and do not take the time to complete these activities, there is valuable information left to be discovered. This information can be used to improve the release process and make it more efficient and cost effective.

The RPM will facilitate many of the activities in the post-implementation phase (see Figure 5.9) and will also capture and document the information gathered. The release cannot be closed until the post-implementation activities have been completed.

Step 7.0 Post-Implementation Review/Lessons Learned

The core of the post-implementation process is to debrief the development process and to understand what could have been done better. In addition to looking for lessons learned about how the release was built and implemented, review of the release process should also be completed. The RPM will schedule feedback sessions with the resources involved with the release development to gather the lessons learned.

© 2010 by Taylor and Francis Group, LLC



Figure 5.9 Post-Implementation phase.

The information will be documented and shared with the RM. The data captured may become the genesis of a service improvement plan, and it is therefore important that the data captured is as precise as possible. The lessons learned document will be stored in the release repository for future reference.

Step 7.1 Complete Release Record

This activity may seem rather basic, however it is oftentimes overlooked, simply due to an "out of sight, out of mind" mentality. Resources involved with the development and implementation of releases are all very busy, and once their activities have been completed, they are on to the next activity on the list.

Completing the release record entails ensuring that all release artifacts and documents are stored in the central release repository. This repository can either be a virtual document repository or a physical repository. Regardless of the type of repository, it should be able to be accessed for reusability of the contents. Release management repositories should be constructed with a service-based nomenclature.

Figure 5.10 illustrates a simple release repository. The IT service of Customer Relations Management (CRM) has a direct relationship to the business of customer service. The releases of the CRM IT service are subfolders of the master IT service. Using this type of nomenclature will enable easy reference of release artifacts for the

© 2010 by Taylor and Francis Group, LLC

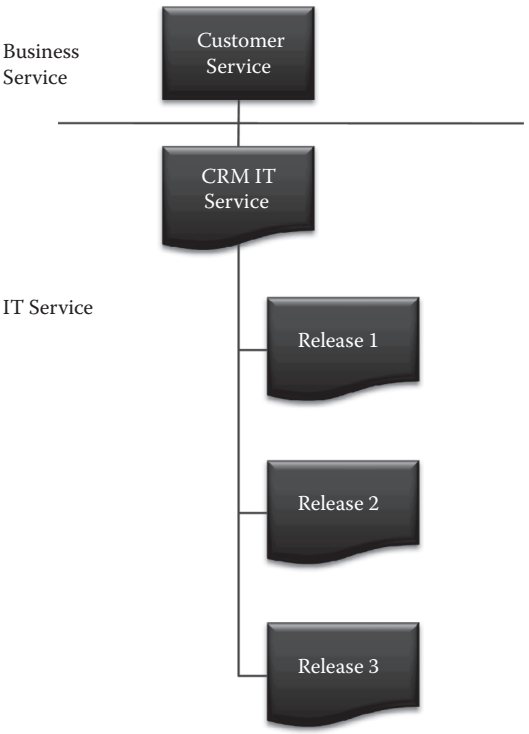


Figure 5.10 Release repository.

master IT service. Some organizations will organize their repository using a project-based nomenclature. While this serves the purpose during the development of the project, it proves difficult to reference artifacts related to a specific IT service. The efficiency gain through reusability is much more difficult to realize due to increased difficulty in being able to reference the relevant artifacts.

The RPM is accountable to ensure that all agreed upon release-related deliverables have been completed, approved, and stored in the release repository. Once all of the artifacts have been stored in the release repository, the release is referred to the RM for review.

Step 7.2 Measure KPIs and Report

The next chapter is devoted to release metrics and what to measure. This step of the post-implementation phase is a step in the RLC that should be completed for every release. The key performance indicators (KPI) measured for the individual release will provide transparency to the organization on the success of the release. These KPIs will translate to performance reports used by the service owners to determine

© 2010 by Taylor and Francis Group, LLC

the success and areas for improvement of future releases. Specific types of reports will be discussed in the next chapter.

The initial set of reports should be facilitated by the RPM to ensure the reports generated match the needs of the service owner. The RM will be more concerned with release process performance reporting. More information can be found in Chapter 6.

Step 7.3 Close Release

It has been a long road, the release has been implemented, the documentation has been completed and approved, and the time has come to close the release. The RPM will forward to the RM a request to close the release. The RM will review the request and ensure all requirements of the release have been met. Once the RM is satisfied that the requirements have been fulfilled, a meeting with the Release Control Board (RCB) will be scheduled to review the results of the release and the lessons learned. The RCB will review the package presented by the RM and approve the release for closure. Depending on the mechanism being used by the organization to track releases, the RM will close the release record.

Benefits

The benefits of a post-implementation review are often overlooked. Many times the accepted approach is "once the release is implemented, the work is over, move on to the next task." The real benefit from taking a step back and reflecting on the release is to understand what went well and what could be improved. These lessons learned can provide great insight and be the basis for continual improvement of both the release process itself and the development practices and methodology for the respective service or application.

Being able to archive release artifacts in an organized manner increases their future usage, increases efficiency, and reduces development cost through reusability. Other benefits of post-implementation review include:

- Process and release reports
- Identification of lessons learned and areas for improvement
- Measurement of KPIs to indicate effectiveness and efficiency of process

Artifacts/Documents

The artifacts used during the post-implementation phase are facilitated by the RPM and used for understanding what went well and where there are opportunities for improvement. Possible improvements in the archiving of artifacts and documents for optimal use should be discussed during this process. The use of this approach increases reusability.

© 2010 by Taylor and Francis Group, LLC

Document	*Who Provides*	*Description*
Post-implementation review (PIR; see Appendix N)	RPM/RM	Provides a guide to review activities of the release after implementation. Creates a place to document lessons learned and is a reference document for future releases.

Using the RLC

Several concepts have been discussed during the detailed discussion of the RLC, and within this discussion there were many recurring concepts: quality, reusability, improving efficiency, and time to delivery. All of these are important benefits of using a structured release process. In the beginning of the chapter, implementation considerations were discussed. One critical concept that was not covered was right sizing the release process to the size of the organization or the size of the release. Both are critical to the success of implementing an effective release process.

A second part of increasing efficiency is related to understanding how the RLC aligns with other developmental processes and with other service management processes. Being able to utilize and share deliverables increases the efficiency of both the release process and other processes. The alignment is not only with project management but with other service management processes.

Right Sizing

There are two aspects of right sizing. The first is right sizing for the type of release and the second is right sizing for the organization. Right sizing is fitting the release process and deliverables to the maturity of the organization and the complexity of the release. Both scenarios need to be considered when implementing the release process.

One concern that organizations with smaller IT operations have is whether there is enough staff to properly operate a release process. This is a valid concern if the approach is to assign dedicated resources to perform specific tasks. However, if these organizations design, build, and implement IT services into an environment managed by the organization, then there is a need for a release process. The concept of implementing is specifically called out since smaller organizations may either purchase or integrate software or services into their own environment. In this instance, development of the service is done by another company, however there are

© 2010 by Taylor and Francis Group, LLC

integration activities that need to be done to bring the purchased components into the organization's environment.

There is no doubt that smaller organizations can gain from the benefits of a release process; understanding the objectives of the process and how to achieve those objectives is the key. Any size organization can use the RLC; the important consideration is to assess the activities essential to the development and integration goals. Additionally, resources can share roles within the organization. For example, one resource would fill the change manager and release manager role. Another example would be a situation where the application owner could also be the release project manager. The benefit of planning, testing, and scheduling of releases can be realized as long as organizational alignment and vision can be obtained.

The second consideration of right sizing is right sizing for the type of release; complexity and maturity are attributes of a release that need to be understood. During the initial release planning and strategy session, a clear understanding of the size and complexity of the release needs to be gained. Based on this assessment, a discussion can take place between the development team and the release manager to determine what artifacts and deliverables will be required for the release. A written agreement between the delivery manger and release management should be completed. This agreement then becomes the basis for the release plan and will be used by the RPM to manage the deliverables. During the discussion of deliverables, great care should be taken to ensure that only the necessary deliverables are identified and agreed upon. Two examples of this approach are:

- The organization determines that a standard word processing program should be purchased; since the software is an industry standard, system testing and user acceptance testing does not need to be completed. However, since the organization uses various desktop configurations, the word processing program needs to be tested on the various configurations. An integration plan, a training plan, and an implementation strategy need to be completed. In this scenario there would not be any need to complete the NER process since an environment is not needed to run the software. However, activities in the operational readiness subprocess need to be completed.
- The organization decides to replace its core receivables system. This is a multiyear project and will consist of several releases. An initial meeting is held with the program manager to determine the strategy and approach. The program will include building a development, test, stage, training, and production environment. The code and software are being developed in partnership with a major software company. Full integration testing will need to be completed. Since the core receivables system is used by a majority of users in the organization, a significant training effort will be needed. Release management will need to be involved to assist the development team and ensure that a quality release is delivered.

© 2010 by Taylor and Francis Group, LLC

Right sizing the deliverables will eliminate unnecessary time and effort of the development team creating non-value-added work products, subsequently reducing development cost.

Service Management Alignment

Release management is an integral part of the service management lifecycle. Even though the process resides in the Service Transition function of the service management lifecycle, there are connections throughout the lifecycle. The relationship between release and the other core processes strengthens the entire service delivery function.

The alignment with the service management lifecycle (see Figure 5.11) binds together the holistic attributes of release management and blends together the aspects of strategy, alignment, and sustainment. Creating the strategy, designing, transitioning, and operating the release brings together the central strategies to provide alignment with the organizational strategies. Sustaining the operational efficiencies of the release using the plan, do, check, act (PDCA) approach drives the continual improvement of the service.

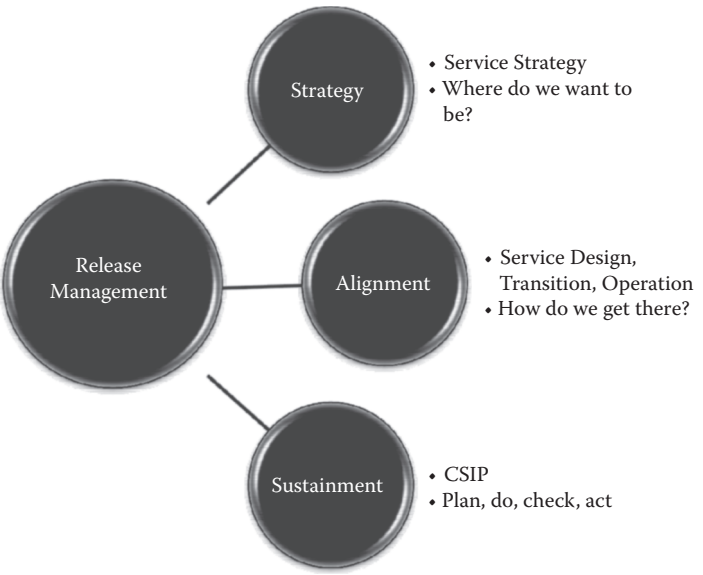


Figure 5.11 Service management alignment.

Change Management

Release and change management have the greatest dependency on each other. In fact, in some service management tools, the release function is part of the change management module. The relationship is strong since a release can be made up of one

© 2010 by Taylor and Francis Group, LLC

or many changes. Ensuring proper testing and release acceptance are done within the RLC. These two aspects of the RLC are key inputs into change and ensuring release readiness is the basis for approval by change management. Documentation of testing and identification of defects are critical to the proof of quality the change approval mechanism is looking for to support the approval of the change. Other inputs from the RLC are the RFCs raised during the different subprocesses, and the activities of the RLC create the artifacts needed to support the execution of the change. Conversely, the change process governs the changes made to enable the release.

Change management also plays a key role in the release management process by helping to ensure that scheduling of the release is completed in an expeditious yet controlled manner. All phases and processes of the RLC lead to the RFC and the approval of the change. Change management also helps to ensure that changed CIs are documented in the CMDB.

Although release management oversees the details of the implementation of a change, it remains under the control and authority of change management.

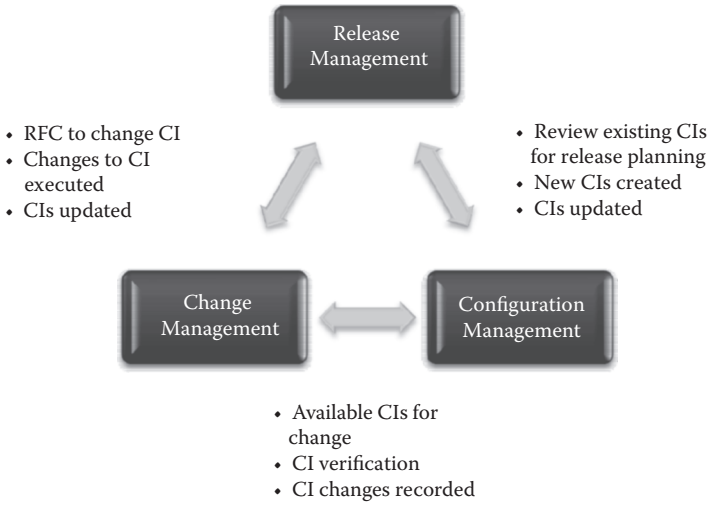


Figure 5.12 Release, change, and configuration relationship.

Configuration Management

Before a release can be developed, there should be an understanding of the existing application, system, the associated CIs, and the relationship to other release units. This information is stored in the CMDB and managed by configuration management. As the new release is developed, configuration of the release is captured and documented through the RLC process. This information is subsequently used to populate the CMDB.

© 2010 by Taylor and Francis Group, LLC

As mentioned in the change management relationship section above, release, change, and configuration have very close relationships and dependencies; Figure 5.12 illustrates the relationship between the three. Procedures need to be put in place to ensure that CIs created or modified during the release process are updated in the CMDB. At the same time, timely updates to the CMDB must be done by configuration management to ensure the CIs are available so that change management can execute changes in them. Change management must also confirm that changes made to a CI are properly recorded in the CMDB so that when the next release is created, the CIs are up to date and accurate.

The ownership of the definitive media library (DML) resides with release management; the administration of the technology resides with configuration management. Updates to the DML are done throughout the RLC by the release manager as the development of the release progresses. The "gold" copy of the existing software is always maintained in the DML and is never replaced.

Availability and Capacity Management

The planning and strategy design phase of the release utilizes information from availability and capacity plans to help create the release. Standards established in both plans need to be used as a reference and guide on how to provide the service's availability and capacity requirements. As the release moves into production, understanding how availability and capacity monitoring and measurements can be used to improve the delivery of the service. This information can be used to create service improvement plans.

Incident Management/Service Desk

During the RLC, support and escalation documentation for the release is created or updated. The support documentation should include creating and resetting passwords, support scripts used to diagnosis symptoms of a loss or degraded service, and escalation procedures. The support documentation needs to be provided to the service desk and incident management for review and understanding. The information will need to be uploaded and stored in the organization's knowledge repository. Additionally, during the testing phase of the RLC, known errors may be discovered and documented in the known error database (KEDB); if there is a workaround, this is also documented in the service desk knowledge base.

Incident management and the service desk will need to be involved in early life support of the release. The involvement may be minimal or they may play an important role depending on how the early life support process is designed. If a major incident outage or incident occurs after release implementation and within the warranty period, incident management would definitely need to be involved. Because

© 2010 by Taylor and Francis Group, LLC

of this involvement, the RPM must ensure that the expectations and roles played by incident management and the service desk are reviewed and understood.

Problem Management

The benefit of doing various types of testing is to unveil any issues or errors within the release; these errors are assessed for risk and analysis is done to determine whether the errors should be corrected or accepted and corrected in a future release. If the latter is true, a workaround should be established to address the known error when it occurs. Documented known errors and associated workarounds will help to resolve incidents and are used to assist problem management in completing root cause analysis. Documentation of test results and operational readiness information can also be reviewed during the problem resolution process and used for further root cause analysis and known error identification.

Service Level Management

In the development of the release plan and strategy, system availability and performance requirements, support service levels, and maintenance windows are determined by the customer. These considerations need to be understood so the proper sizing and architecture of the release can be designed and built to meet the desired requirements. These requirements, articulated by the customer, must be captured and documented as service level requirements. The service level requirements are the basis for the creation of the service level agreement that will be reviewed and agreed upon by the customer. Service level management maintains the baseline availability and service levels as well as the schedules and costs associated with nonbaseline variables.

Service level management will also use these baseline requirements as a basis for creation of a service offering. The service offering will be used to create the service catalog.

IT Service Continuity

Planning for service restoration usually takes a backseat to other development and release activities. Planning for and creating a service restoration plan for the service is like buying an insurance policy—no one likes to pay for the insurance policy, however when a catastrophic event occurs, you're glad you have the policy. The same concept applies to preparing for a catastrophic loss of service. The best time to start preparing is during the creation of the service. This requirement is fulfilled during the OR process with the requirement of providing the service restoration plan. This requirement needs to remain in place whether the release being developed is a new release or an upgrade release. The service restoration plan becomes part of the overall IT enterprise service continuity plan.

© 2010 by Taylor and Francis Group, LLC

Continual Service Improvement

It is sometimes hard to understand how Continual Service Improvement (CSI) can be related to a process that deals with the creation or modification of a service, however CSI can be the genesis of the creation of a new release. CSI deals with continually striving to improve the performance and quality of IT services. CSI uses the PDCA approach to create opportunity for improvement, which usually results in the initiation of a new release.

IT Security Management

Managing IT security has become a primary concern in delivering IT services. There are many ways to approach security and ensure the proper controls are in place. One of the easiest ways to incorporate security mechanisms and monitoring is to include them when the service is being created. Building for security starts with the requirements and design of the service. During the different phases of the RLC, security requirements need to be reviewed to ensure they are being met. If the organization has a group responsible for IT security, release management will need to work closely with the group to ensure security standards are being met. A final security review of the release can become part of the final OR process if so desired.

Knowledge Management

When a release is being created, the thought of knowledge management is not normally considered, however the artifacts created during the development of the release are now part of the knowledge base of the organization. A primary benefit of a release practice is reusability and improved service operation, and the way these benefits are realized are through documented knowledge capital. Therefore it is paramount to capture the knowledge being created in some type of knowledge management database (KMDB). The KMDB may be a centralized document repository or knowledge management tool with full search capabilities.

Procurement Management

Every organization has some type of procurement process, whether it is centralized for the entire organization or a dedicated function within the IT space. Procurement management as it relates to release management interfaces with the activity versus the role. The release process provides a mechanism to ensure components that need to be procured are architected and designed to realize maximum value and the most favorable cost. Being able to clearly document the requirements of the release provides accurate procurement requirements. The interrelationship mainly comes out of the NER process.

© 2010 by Taylor and Francis Group, LLC

Concept Review

The release lifecycle is the heart of the release management process. It is a structured, repeatable framework increasing the efficiency, quality, and reducing the time to delivery of releases. The RLC is made up of seven subprocesses that take a release from planning and design to the implementation and post-implementation of the release. Utilization of the RLC creates consistent reusable processes and deliverables.

- The benefits of the RLC are related to efficiency, quality, and financial considerations, and these benefits are realized through consistent, repeatable, and sustainable processes.
- The RLC defines activities and tasks that will facilitate a quality release that is reliable, maintainable, and serviceable, meets the requirements of the customer, and enables the business to achieve its goals and objectives.
- Understanding implementation considerations is a necessity. Consideration of organizational maturity and culture needs to have an influence on the implementation method.
- The RLC is made up of seven subprocesses: Initiation, Release Configuration and Build, New Environment Request, Quality Assurance, Operational Readiness, Implementation, and post-implementation. Each of the processes has distinct goals and objectives.
- Even though the seven subprocesses complement each other, many of the activities of the subprocesses are completed in parallel with each other. The RLC is not a sequential framework even though there are dependencies.
- A key consideration when implementing a release process is right sizing the process and deliverables to the size and need of the organization and release. A smaller organization will need to share process roles and responsibilities and assess for value-added processes. A minor release may not need all of the same release deliverables that a more complex release needs.
- Release management is part of the Service Transition function within the service management lifecycle; however, it has influences and ties to all of the core service management functions.

© 2010 by Taylor and Francis Group, LLC

Chapter 6

Release Measures and Metrics

There is an old saying: "What is measured is managed, and what is not managed is assuredly out of control." This saying captures a simple concept—being able to document and measure the current state creates an understanding and provides a basis for assessment, improvement, and opportunity. Without this knowledge, improvement and growth will either be slow or nonexistent. While improvement is important, a more tangible benefit that is more important to the organization is the value that is being realized from the process—whether it is efficiency or quality, both result in financial ramifications. Being able to demonstrate the financial value of release management is critical in gaining acceptance of the process.

Setting the Baseline

A basic concept of creating a measurement environment is being able to define a starting point. Without a starting point, there is no basis for understanding how effective or ineffective a process is. This starting point is called a *baseline*. A baseline is a measurement of the current state, whether it is how fast widgets are being pushed through the manufacturing process or the number of errors that occur.

Organizations lacking a measurement system ask where to start, and offer comments such as, "Our processes are so ineffective that there is no need to measure the 'as is'; we wouldn't know how or what to measure." This may be the situation, but

© 2010 by Taylor and Francis Group, LLC

to create an effective measurement process, the existing ineffective process needs to be measured. The single most important reason to do a baseline measurement is to provide a way to demonstrate the improvement the new process has made. Being able to provide documented evidence of improvement is paramount to gaining continued support and funding for the program.

Setting a baseline starts with understanding what needs to be measured, and great care must be taken not to try to measure everything. This will yield a lot of data but no understanding of what one is trying to measure or how to use the data when collected. When starting to determine what should be measured, an organization should start with a small number of high-level value-added metrics. Once the organization begins the process of determining what metrics should be measured, the normal tendency is to continue to increase the scope until the metrics being measured are unmanageable and become a burden to gather. A recommended approach is to select three to five areas within the process that are perceived to be trouble areas. Determine whether there is value in measuring these areas; then create a metric that will provide data that can be used to take action to improve the process. Once the three to five metrics have been agreed upon, determine at what interval the measurements will be taken. Since the metrics are high level, it normally will take at least a month's worth of data to determine if there has been any change. Later in this chapter a discussion of specifically what should be measured and why will be provided.

Once the metrics to be measured have been agreed upon, a baseline measurement of the metrics must be developed. Determining the mechanism for capturing and reporting on the metrics will be used to take the initial measurements; this initial measurement will become the baseline against which the subsequent data gathered will be measured and managed. As previously mentioned, it is important that the initial measurement or baseline is taken before any improvement or process enhancements take place. This will demonstrate the success of the improvements and provide a tool to be used to show the value being added.

Value of Measurement

Developing metrics and measurements for the sake of saying the process is being measured is almost as beneficial as not having any metrics; in some instances it can be worse. Developing reporting that is not evaluated and used is time consuming and costly. It is more beneficial to the organization to develop fewer metrics that are value added and actionable than producing many metrics that offer no value.

The value of metrics should be measured against how actionable the measurement is; can some action be taken as a direct result of the data provided by the metric? There is not always a direct one-to-one correlation of metric result and action; the metric may be one indicator coupled with another metric to indicate

© 2010 by Taylor and Francis Group, LLC

that some action needs to be executed. An example is the following: When reviewing monthly release implementation reporting, the metric of the number of defects in releases continues to increase. This number alone raises some suspicions, however, the actual effect of the trend is not clear. During the same review, it is noted that the metric for releases causing incidents has also increased. Investigation is done and it is determined that recent releases implemented are causing increased service interruptions, increased downtime, and increased calls to the service desk. Tying the various metrics together, it is determined that there must be an issue with the testing process. As a result, a service improvement plan is created to improve the testing process and review.

If a metric can be quantified and associated with a financial cost or measure, it is even more value added. A simple example of this type of metric would be the time it takes to complete the implementation process of the release. The time measurement can be associated to the cost per hour of the resources needed to complete the implementation. The formula would be:

of implementation hours × # of resources utilized × hourly rate = cost of implementation

This is a rather simplistic measurement, but there are many organizations which if asked, could not provide this information. This information can also be used to actually bill the customer for the implementation of the release, to look for ways to improve processes and reduce costs. There really is no measurement more valuable to an executive than understanding the costs of activities and processes. Providing this information is very actionable and directly affects the bottom line of the organization.

Performance management is another aspect of value-added metrics. The individual, team, department, and organization's performance can be evaluated through the correct measurements. Metrics such as the number of defects in a release is a direct measurement of the performance of the development staff.

A review cycle of metrics should be established as part of the continual service improvement process to ensure that the metrics being measured still add value to the organization. A metric may be established when a new process is implemented to identify the success of the process; however, after the process matures, there may no longer be a need to report on that metric. An example of a metric established at the beginning of a new release process is the number of releases being implemented without an IT service continuity plan. This may be a common practice before the operational readiness process was implemented, but once the process is in place and the practice of having a continuity plan is assimilated and is in common practice, the metric ceases to add value. Therefore, that metric should either be modified or eliminated.

Most organizations have some type of metrics in place. They may not be directly tied to the release process if a process is not established, however there will be some

© 2010 by Taylor and Francis Group, LLC

types of metrics. A simple test should be done of those metrics to determine if the metrics add value. This simple test consists of three questions:

- What value does the metric provide?
- Is the metric actionable?
- Who uses the metric and what is it used for?

If the answers to these questions cannot be articulated, then the metric is not value added and should be considered for retirement. In established organizations, there may be reporting being done that has not been used for years and does not have any relevance to today's operation, however, it continues because the reporting is automated. Even though this type of reporting is self-sustaining, it still has a cost of processing and possibly printing. How many times does an organization have reports that print every morning, and are then either filed or thrown into the recycle bin without anyone looking at them? When the person doing the morning activity is asked why is this being done, and the answer is "because it has always been done this way," it is time to ask the three simple questions to determine the value of the metrics.

Components of Metrics

Creating metrics and measurements is one of the biggest challenges organizations have, and this does not even consider the creation of value-added metrics. This is especially true for release management. In a process that crosses so many verticals, the question becomes, "What should we measure and how much should we measure?" Many organizations struggle to understand the purpose of metrics and how the metrics should be used. Metrics can be classified in three categories: operational, efficiency, and quality. Each of these types of metrics have their own characteristics and purpose, but they are interrelated and dependent on each other in some instances.

- **Operational metrics**—These types of metrics are one dimensional, straightforward, and are usually the easiest to be gathered. They are generally number driven and if calculations are used to report on them, the calculation is usually simple and straightforward. Examples of operational metrics are the number of releases in process or number of failed releases.
- **Efficiency metrics**—Metrics that measure efficiency can also be said to measure productivity. These types of metrics are multidimensional and need some type of calculation to present the results. Efficiency metrics usually use operational metrics as one denominator to produce the final output. As the name indicates, efficiency metrics provide measurements of effectiveness of the process or attribute being measured. The reported metric is usually in

© 2010 by Taylor and Francis Group, LLC

percentage form. Efficiency metrics may need additional explanation for the user to interpret the metric. Examples of efficiency metrics would be the percentage of failed releases or the percentage of minor defects accepted prior to release implementation.

- **Quality metrics**—The measurement of quality can be straightforward or it can be tricky. Much like the efficiency metric, the quality metric is also multidimensional. Typically, some type of calculation needs to be used to present the metric. Quality metrics can use operational metrics as well as efficiency metrics in the calculations. The availability, reliability, and maintainability of releases are components that will be used in calculating quality measures. While some types of quality measurements will provide direct indicators of customer satisfaction, others will have an indirect effect. Quality metrics can be presented either in percentage or numeric format. Examples of quality metrics are the percentage of service unavailability resulting from a new release, the number of unknown errors discovered after release implementation, and the percentage of customer satisfaction with the new release.

Once the foundation of creating and understanding metrics has been established, the next question is how to make measurements meaningful and value added for the organization. This is done by creating key performance indicators (KPIs). Earlier it was mentioned that the organization needs to understand what goals are important and what it needs to measure to achieve these goals. KPIs should be focused on supplying the information needed to determine if these goals are being achieved. In the same way that quality measures may not have a direct link to customer satisfaction, they most assuredly need to be business driven.

KPIs normally do not exist as independent measures, and are normally the product of a combination of one or more operational, efficiency, or quality metrics. The KPI can be as simple as the number of releases implemented in a thirty-day period or as complex as the percentage of UNIX programmers used to develop the release and the total cost for those programmers. It should be kept in mind that the more complex the KPI, the tougher it will be to measure and manage, however the organization's KPIs should be as simple or as complex as necessary to achieve the level of desired control and management. The rule with KPIs is "less is more," so create and choose KPIs that are meaningful and add value.

Figure 6.1 illustrates a structured approach to creating metrics than can be used to measure the performance of a process or service. Using this methodology provides a mechanism to relate business objectives and strategies to the performance of processes and services. Building on the basis of measurements and KPIs, critical success factors (CSF) tie the metrics directly to the business relationships.

Critical success factors represent operational performance results of processes and services provided by IT to the business. In the same way that KPIs are derived from one or more measurements, CSFs are made up of one or more KPIs. When

© 2010 by Taylor and Francis Group, LLC

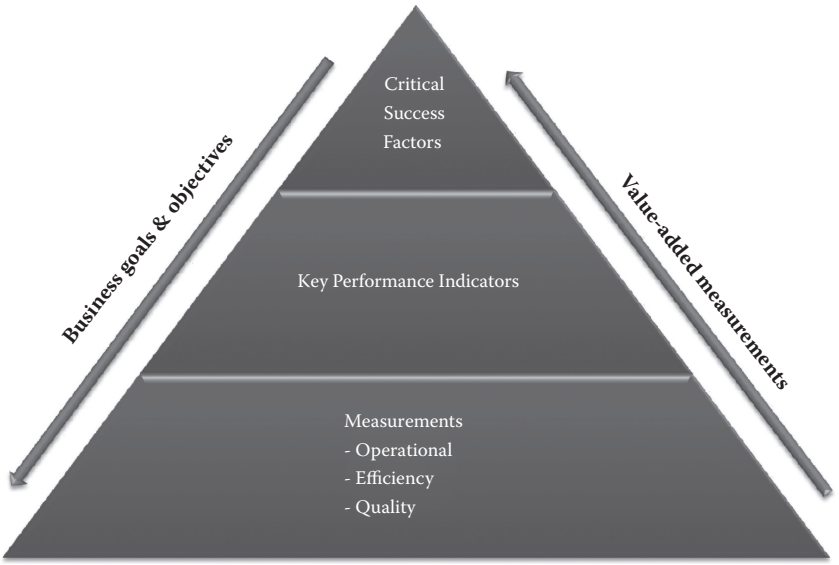


Figure 6.1 Measurement creation.

creating CSFs, the measurement should be related to a business objective that can be measured by one or more KPIs. The CSF will normally be created in a more narrative form rather than a quantitative form. An example of a CSF for the release process would be "Implement quality releases." The CSF is narrative; the reporting measurement would be "the percentage of quality releases delivered in the last six months." CSFs are normally created at a leadership level, and the KPIs and measurements support the achievement of the CSF.

Understanding the components of measurements and the framework of how metrics are created provides the basis for designing measurements for the release process. Using the measurement framework will ensure that the metrics being created add value and help the organization to achieve its objectives.

Creating measurements for the release process can be difficult, and determining what should be measured can be confusing. When creating the measurements and subsequently the metrics for the release process, using a simple test of the three questions previously mentioned is a good starting point.

- What value does the metric provide?
- Is the metric actionable?
- Who uses the metric and what is it used for?

A common pitfall when creating metrics is to try to measure all facets of the process. Keep in mind that the best approach is to create metrics that demonstrate the value of the release process. Being able to provide measurable results is a key

© 2010 by Taylor and Francis Group, LLC

to garnering support for the process development and gaining the resources and financial support of executives. Remember, less is more.

Measurement and Metrics

Thus far in the chapter a baseline understanding has been set of the importance of measuring and reporting. An understanding of how metrics are created has also been discussed. So the next question normally is, "What specific metrics and measurements should be used for release management?" Before examples of measurements can be provided, an additional understanding is necessary. When deciding on measurements, there are two categories to be considered. The first is *process measurements* and the second is *measurements of the specific release delivery*.

- **Process measurements** provide information on the efficiency and quality of the activities of the process. Examples would be average time for delivery of an environment using the New Environment Request (NER) process or the maturity of the release process.
- **Release delivery measurements** provide information on the efficiency and quality of a specific release or group of releases. Examples of this type of measurement are the percentage of defects for either the specific release or for a group of releases.

Being able to create both types of measurements will provide the information to identify performance and specific areas for continual improvement. Once this information is gathered and analyzed, a service improvement plan can be created and executed.

Table 6.1 includes suggested metrics that can be used when creating release measurements. Using the framework discussed earlier, the measurements will be built from measurements based on KPIs to CSFs.

Capturing these metrics can be done either manually or through data gathered using a service management tool. Reporting of these metrics is important to the development team, release manager, and the manager of the service being implemented. The reporting time period should be established as a base requirement. The reporting time may vary depending on the type of measurement.

Key performance indicators (Table 6.2) are a gauge for determining efficiency and can be a good indicator of areas for improvement. KPIs can also provide clarity of best practices that should be emulated within the development teams.

Critical success factors (Table 6.3) tie the measurements and KPIs to the value proposition of the process. CSFs need to be able to demonstrate clearly what the benefit of the process is, without being able to provide evidence, justification of resource allocation, and funding for the process or activities will be difficult.

© 2010 by Taylor and Francis Group, LLC

Table 6.1 Measurements

Ref #	*Type O, E, Q*	*Purpose P, D*	*Metric*
m1	O	P	Number of releases implemented for a time period
m2	Q	D	Number of releases successfully implemented for a time period
m3	Q	P, D	Number of releases implemented causing incidents
m4	E, Q	D	Number of accepted known errors when release is implemented
m5	E	P	Total days for release development
m6	E	P	Number of releases rescheduled
m7	O, Q	D	Number of training questions logged during early life support
m8	O, E	D	Number of test cases mapped to business requirements
m9	Q	D	Number of successful test cases executed

Note: Type = Operational, Efficiency, Quality; Purpose = Process, Delivery

Table 6.2 Key Performance Indicators

Ref #	*KPI*	*Product*
k1	Successful releases for a period of time	m1 – m2 / m1 × 100
k2	Releases causing incidents	m1 / m3 × 100
k3	Average days to implement a release	m5 × m1 / m1
k4	Release efficiency	m6 / m1
k5	Quality of releases being implemented	m4 × m1 / m1 × 100
k6	Release training efficiency	m7 × m1 / m1 × 100
k7	Successful test cases	m8 / m9

© 2010 by Taylor and Francis Group, LLC

Table 6.3 Critical Success Factors

Ref #	*CSF*	*Indicator*
c1	Successful release process producing quality releases	k1, k5
c2	Consistent efficient release process	k4, k6
c3	Release process maps to business needs	k5, k6, k8

The measurements, KPIs, and CSFs created should be tailored to fit the needs of the organization. The examples provided can be used and supplemented to fit the need. The number of measurements and metrics will also differ. It is recommended to start with simple measurements and expand them as the process and the organization mature.

Concept Review

Not being able to determine process performance or measure output leads to inefficiency and wasted financial investment. This is not to say that without measurements process cannot exist and operate, however not establishing metrics and performance measures leads to inconsistent process, reduced quality, and increased rework. This leads to increased downtime and reduced reliability and stability, translating to reduced productivity and reduced revenue generation for the organization. Establishing value-added measurements and metrics leads to increased productivity.

- Understanding the current state is the critical first step in creating measurements. It does not matter what the current state is, but a baseline must be established. Defined measurements may not exist, so measure what can be measured.
- Measurements and metrics must be able to provide value-added data. Creating metrics for the sake of having metrics is a waste of time and resources. To provide value, metrics and measurements must be actionable; what actions can be taken as a result of the data being presented? If the metrics are not actionable, there is no value and the use of the metric should be questioned. Three simple questions can be used to determine if the metric or measurement adds value.
 - What value does the metric provide?
 - Is the metric actionable?
 - Who uses the metric and what is it used for?
- Using a framework to develop measurements and metrics provides a defined methodology that enables a mapping of base measurements through critical success factors. Establishing the relationship increases the value that measurements provide.

© 2010 by Taylor and Francis Group, LLC

- Base measurements can be categorized into three types: operational, efficiency, and quality. Creating a balanced measurement system must include metrics that address each of these types of measurements. Measurements should quantify how the process is operating and also how the delivery of the individual release development process is being delivered.
- Key performance indicators provide a view of significant outcomes desired of the process. These outcomes determine how successfully the process is being performed. Normally, KPIs are a result of a calculation of operational, efficiency, and quality measurements.
- Tying the outcome of the process to business objectives provides clear justification of process value; this is done through the creation of critical success factors. The number of CSFs usually is limited as the focus of CSFs is how the process relates to the objectives of the organization.
- Creating measurements and metrics should begin with simple and easy measurements; as the process and organization matures, so should the measurements.
- The simple premise to remember is: What is measured is managed; what is not measured is assuredly out of control.

© 2010 by Taylor and Francis Group, LLC

Chapter 7

Selling Release Management

Organizations have an easy time accepting the operational aspects of service management; practices such as incident and problem management are easy to understand. Even practices such as change and configuration are easily justified and the benefits of the process are understood. However, when the concept of release management is raised, red flags are raised and the walls go up. There are many reasons for this. The primary reason is a failure to understand what release management is and the benefits derived from a consistent, repeatable process focusing on quality. This can be especially true in organizations that have some type of project management process. In these organizations it is very difficult for managers to recognize and acknowledge the differences between release processes and project management processes. This confusion is understandable; on the surface, both processes look very similar. Chapter 3 outlines the differences between the two processes, and also discusses the similarities and complementary activities between release and project management. Being able to articulate and demonstrate the differences and similarities between the two processes is crucial to gaining acceptance of implementing a release process.

Being able to articulate the benefits of release management is crucial to being able to gain acceptance of the process, however this is just one of the pieces of selling the process to management. Other pieces include understanding who your allies, supporters, and detractors are—knowing how to navigate the perils and influence those who need to be influenced. A final consideration is the readiness of the organization to accept the discipline needed to make a release process successful.

© 2010 by Taylor and Francis Group, LLC

Organizational Readiness

Earlier the concept of organizational readiness and maturity was touched upon. Organizations that try to implement processes prior to being mature enough often fail and can create roadblocks to future implementations. Even if an IT organization has been in operation for a long period of time, the operational maturity level may still not be at the appropriate level. These organizations typically struggle with excessive downtime and do not have direct alignment with the objectives of the organization. Typical characteristics of organizations that will struggle with a release process are organizations not aligned with the organization's strategy. Not only is there a misalignment with the organizational strategy, there is also misalignment within the IT organization or there is no IT strategy. The usual result of not having a defined IT strategy is that individual functional groups use their own processes and address their own agendas. There is no alignment or synergy within the IT organization.

Typically, IT organizations will mature as a structured service delivery program is implemented and executed. The service delivery program typically will start in the areas of incident management and then progress to change management. This is the typical progression as these processes usually have the greatest impact on improving delivery of services. A by-product of these processes is that resources look for and ask for consistent processes that will improve the planning, quality, and delivery of new development.

Every organization can benefit from some level of a release process, and assessing the readiness of the organization will be paramount in determining at what speed the release process should be implemented. It may be determined that the organization is only ready for the implementation of a structured QA process or structured support documentation processes. Typically the implementation of a release schedule is a more advanced concept and one that should be carefully planned prior to implementation. The complexity of implementing a release schedule is that expectations will normally need to be changed with the business customer that is used to having releases implemented on *its* schedule versus a planned release schedule.

A final consideration of organizational readiness is determining how change ready the IT organization is. Change readiness can differ at all levels—executive, senior manager, middle manager and team members. At each level the change readiness can differ and the reasons can also differ. The reasons can range from not understanding the benefits, to budgetary considerations, to threatening the status quo. Each of these is a valid concern that needs to be addressed. Being able to identify these concerns and being able to address each will lead to a successful implementation.

Executive Buy-in

Whether implementing release management or any other initiative, the golden rule for success is securing executive management buy-in and support. Initiatives

© 2010 by Taylor and Francis Group, LLC

usually begin either as a grassroots movement or a planned corporate initiative. Whatever the situation, if executive endorsement is not secured, the initiative will not succeed. There may be initiatives that begin without full endorsement; these initiatives will struggle to exist and will eventually fade away.

There are different levels of executive endorsement—no endorsement, passive endorsement, and active endorsement—and each level has a direct effect on the outcome of the initiative. The initiative can move forward with the latter two types, but the initiative has little hope of succeeding with no endorsement.

- **Passive endorsement.** This type of endorsement allows the initiative to move forward, provides the resources, and does not put up roadblocks. However, the initiative will not be publicly endorsed and may conflicting initiatives may be endorsed. This type of endorsement makes the initiative much tougher to implement and operate. The success of the entire process falls to the initiative's manager and leaves the selling of the process to this manager.
- **Active endorsement.** This is the most desirable type of endorsement. When an executive actively endorses an initiative, the executive comes out publicly and provides validation of the initiative. Resources are allocated and the processes are created and implemented and supported. The executive leaves little doubt within the organization of the endorsement and the expectation that the new process needs to be followed.

While executives are concerned with the overall benefits of release management and other initiatives, their primary concerns are normally with overall efficiency gains, strategy alignment, and increased return on investment. Therefore, when presenting initiatives to executives, the presentations should be focused in these areas. The points already discussed provide information that can be used in presenting release management to address efficiency gains and strategy alignment. Later in this chapter, the return on investment (ROI) associated with release management will be reviewed.

Middle Management

Middle management can be the toughest group within an organization to move from detractors to supporters. As with any group, there will be some who fully support the release management process, there will be others that oppose any efforts, and there will be those on the fence who will wait to see how the initiative progresses. These characteristics are more amplified in middle management than in other groups. This means that greater effort is needed to demonstrate the benefits of release management to this group.

One approach that can be used to move middle managers from detractors to supporters is advocacy. Creating advocates begins with involving middle managers early in the process; get the most influential and most vocal managers involved with the development process. This is not to say the process design is turned over to

© 2010 by Taylor and Francis Group, LLC

the middle managers; instead, input is received from them during the development process. Their experience can be utilized to make the process better and to add cultural influences that will enable increased acceptance of the process. Advocacy is also increased by enlisting the outspoken managers as pilot groups for new processes. As these middle managers become more involved, they move from detractors to supporters. Since these managers have assisted in the development, they now have a stake in the success of the process. This commitment is translated into advocacy and is related to others. The key concept is: don't shy away from the most critical managers; embrace them and make them a part of the process.

A second approach for influencing this group is to do so indirectly, which is done from the top down and from the bottom up. The top–down approach can be effective, however depending on the support and direction provided from executive management, the level of effectiveness can vary. The effectiveness of a bottom-up approach can vary depending on the receptiveness of the middle managers.

Because middle managers are the toughest group to persuade, it will take the most time and effort to win them over. The Chief Information Officer (CIO) may set the strategy, but middle managers make the strategy happen. Middle managers decide how tasks are accomplished, what resources are engaged, and set the tone for their direct reports. Understanding that middle managers have this power and addressing it will increase the acceptance of the process integration.

Grassroots

When implementing a new process, an important concept that can be overlooked is the power of the people executing the process; remember, these are the people that actually make things happen. If anything can be learned from the past, the people executing the process have the ability to either embrace a new process and make it successful or reject the process and make it fail. Being able to reach the people who make the process happen can be tricky, but this can be done in many ways. An unsuccessful approach is to dictate how the process is to be executed and ignore any feedback or input received. Organizations using this approach usually do not recognize that this is not successful, mainly because employees in these organizations perceive that it is not safe to speak up. The typical behavior would be for the resources to smile and agree, walk back to their work area, and proceed to undermine the success of the process. Management then wonders why the new process is not successful.

The grassroots movement within this group of resources has one overarching concern—"What's In It For Me?"' or WIIFM. This is not to say that this group is not concerned about how the process will affect the organization, however if the new process positively affects them personally, there is a much greater chance it will be successful. So the question becomes, how do you create a process experience that is positive for this group? This can be done by using a concept described in Chapter 6—baselining. Earlier, a strategy method was discussed that asked several

© 2010 by Taylor and Francis Group, LLC

questions: Where do we want to be? Where are we now? How are we going to get there? How do we know we got there? Using this approach, create a baseline of the existing environment (where are we now?) using input from the people executing the process. These are the people providing the reality of what is actually happening. Listen and document these people's ideas concerning how the process can be executed more efficiently, what the roadblocks are that they face, and how they can be removed. Once the baseline is created and ideas captured, work on eliminating the roadblocks. The more that can be removed the better the new process will be accepted. Be sure to acknowledge the contributions of these people.

Communication of how the process will make their lives better needs to be a focus of any implementation plan. The benefits should be communicated in many different ways, posters, presentations, e-mails, newsletters, and even blogs if the organization has one. Take a lesson from the marketing professionals—in today's world, the benefits of a product are presented in many ways. Ensuring that team members understand the new process and have a firm belief in the benefits of the WIIFM will make the process successful.

Bubble-Up Methodology

Three different groups within the organization have been identified—executive management, middle managers, and the executers. The question becomes how to use these groups together to be successful. Earlier in this chapter it was put forth that middle managers can be the biggest supporters or the strongest detractors to the implementation of the release process. Which way they lean can determine whether the process is a success or a failure. The middle managers are usually the toughest group to persuade.

An approach that can be used to influence middle managers to support the release process is the bubble-up methodology (Figure 7.1). This methodology builds

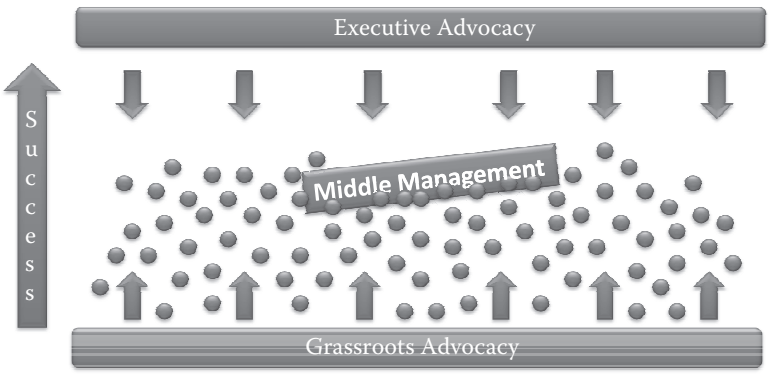


Figure 7.1 Bubble-up methodology.

© 2010 by Taylor and Francis Group, LLC

on the power of grassroots approval and advocacy of the process being implemented and the advocacy of top-down support.

Executive advocacy can be enough to push middle managers to become supporters of the release process, especially if performance objectives and compensation are tied to its success. However, this is not usually the situation. Typical executive advocacy is support of the process at various levels. This allows middle managers the option to determine how the process is adopted. Absent the executive direction, turning to the grassroots advocacy of the resources who execute the process adds pressure for middle managers to accept the release process.

In actuality, the grassroots groups can persuade middle managers more toward advocacy of the process than either executive direction or the process implementation team. Simply, the grassroots group can bubble-up the positive benefits of the process, and the positive benefits in turn result in improved efficiency and increased productivity. Both of these are important to middle managers as they allow them to meet their own objectives, which in turn makes them look good to executive management.

Bubbling up can take different avenues; the easiest to recognize is when team leads and developers speak positively of the process and managers take notice of comments made by their team. Another way to recognize the quality of releases is to improve the time it takes to complete tasks. A third indication of positive bubbling up is the morale of the team. As the process improves and tasks become easier, morale will increase.

From the perspective of the implementation process, this means that special attention needs to be given to the grassroots groups to ensure the WIIFMs are addressed. It also means that open communication and transparency need to be maintained. Being able to create a positive perception will result in positive bubbling up, which will influence middle managers into positive advocacy.

Return on Investment

Executives commonly ask, "What is the ROI of the process? What benefits does the organization derive from implementing release management?" Being able to provide projections of ROI will provide executives with the data to justify the implementation of a release process. The ROI to support a release process can be classified into two areas; the first is in development and second is in the operation of the service that the release supports.

Using a consistent release process will reduce development costs and increase the ROI of the development process. Development process improvements resulting in improved ROI will result in improved quality and time to delivery. ROI improvement areas within the development process are shown in Table 7.1.

Table 7.1 shows examples of ROI value considerations and the calculations that can be used to provide a dollar value or productivity value that can be used to demonstrate the value of release management. These are important considerations

© 2010 by Taylor and Francis Group, LLC

Table 7.1 ROI Improvement Areas in the Development Process

ROI Value	*ROI Calculation*	*Development Process(D)/ Operators(O)*
Reduced time to delivery; savings in resource cost as a result of reusability and consistent processes.	Current avg # of hours for delivery – predicted avg # of hours for delivery (post-implementation) = # hours saved × avg hourly resource cost = resource savings	D
Improved quality; improved requirements mapping to test results.	Current avg # test cases with direct mapping to requirements – avg # test cases with direct mapping (post-implementation / avg # of pre-implementation test cases × 100 = percentage improvement of defined test cases	D
Artifact reusability; savings in resource cost for duplicating release artifacts.	Current avg time to develop a specific artifact × avg resource cost = cost of developing a specific artifact. Cost savings of each artifact not being duplicated.	D
Reduced time to deliver a new environment; improved time to deliver an environment for a release.	Current avg time to deliver an environment – projected improved time to deliver = improved time for delivery × avg cost of infrastructure resource = improved cost of delivery of a new environment	D
Improved service availability; improved uptime of the service as a result of better planning and quality.	Current service availability % – projected service availability after release = increased % of availability	O
Improved service maintainability; improved time to restore service after a service interruption due to documented and accepted restoration plan.	Current avg time to restore service – projected improvement in avg time to restore service = improved restoration time × avg operation resource cost = savings due to improved restoration plans	O

© 2010 by Taylor and Francis Group, LLC

as executive management is interested in what the return is for their investment in dollars and resource allocation. As the calculations are reviewed, it is apparent that measurements of the existing situation are needed. This can be an obstacle, however it can be overcome. Once a decision has been made regarding the ROI calculations to be used, the baseline calculations will need to be gathered; this may be an easy task or a complicated task depending on the willingness and openness of the organizational resources.

Release Management and the Business

Thus far the primary discussion has focused on the benefits and selling of release management to the various resources within IT. Unlike other service management processes, release management directly affects the delivery of services to the business. The business customer can be the most critical and most demanding of any group connected with the release process. The criticality comes from the high expectation of delivery of a quality service that will improve efficiency and in turn increase revenue. This high level of expectation is compounded by the desire to have the product "yesterday," and the idea of having to wait until the next scheduled release is not acceptable.

Much like selling the benefits of release management to the other groups, the focus needs to be on the WIIFM—What's In It For Me? This begins with providing the benefits of increased quality that translates to increased availability of the service. Demonstrating to the business how the requirements put forth by them are directly related to the testing of the release will increase their confidence in the process. Typically, organizations that do not have a release process will deliver system enhancements as quickly as possible. This usually means shortcuts are taken that translate into a less stable service and reduced availability. This simple sales pitch can be used to gain the business's approval: "Using the release process, the IT service or enhancements requested will be delivered on a scheduled date, without delays, and the stability, maintainability, and reliability of the service will be increased." The increase of quality and on-time delivery will normally be enough for the business to give the release process a try.

Providing the carrot of increased quality and improved delivery times may be enough, however using the ROI calculations discussed previously can help sell the benefits. Additionally, doing a voice-of-the-customer survey will help to understand the pain points of the business and can provide insight into areas that can be addressed during the discussions of the benefits of release management.

Concept Review

Being able to create an effective process and execute that process can be a challenging and rewarding endeavor. However, before these efforts can begin, approval

© 2010 by Taylor and Francis Group, LLC

needs to be obtained. Once the approval is obtained, the planning and development can begin. One of the most difficult pieces of the implementation will be the cultural acceptance of the new process, and the cultural aspects and getting the necessary support needs to be part of the implementation strategy. In this chapter the different aspects of how to sell release management to the various parts of the organization were discussed. Ideas concerning how to gain approval and moving people from detractors to supporters were explored.

- The primary concept to be considered before embarking on the implementation of a release process is organizational readiness. Is the organization mature enough in the existing processes to appreciate and accept the rigor that a release process brings? A second consideration is how ready the organization is for change. Does the organization embrace change or is it rejected? Understanding these aspects of the organization will determine the level and speed of implementation.
- The golden rule of a successful initiative is executive buy-in and endorsement. Without public executive endorsement, the initiative is destined to be an uphill battle for acceptance within the organization. There are two types of executive buy-in and endorsement—passive endorsement and active endorsement.
- Middle management can be the toughest group in an organization to move from detractors to supporters. This group can make or break the release management initiative. There are three behaviors exhibited by middle managers—detractors, supporters, and the agonistic. Each of the behaviors must be recognized and the strategy needs to include how each of these will be addressed. Techniques to move detractors and agonistics to supporters include advocacy and influence, both direct and indirect.
- The third group of people to be considered is the people who execute the process and activities. This group of people has tremendous influence on the middle managers and can be a tremendous asset in influencing those in management. Creating grassroots support is essential to the success of the initiative. A key to gaining the support of this group is to focus on the WIIFM—What's In It For Me? How the release process can make the executers' task more efficient and meaningful needs to be demonstrated to gain their support. For this group, it is not enough to provide the benefits for the organization; the benefits to the individual must be clearly understood.
- Moving middle managers to advocacy can be done by using the bubble-up methodology. Influencing from the top down and bubbling up support from the people that execute the process moves the middle toward advocacy.
- In today's environment the focus of all organizations is value and the return on the investment that is being made. The pressure on executives to demonstrate a high level of ROI comes directly from the shareholders; therefore it is a primary

© 2010 by Taylor and Francis Group, LLC

concern. Release management provides a process that will improve the ROI of the development and release process, and the ability to articulate and provide projections of improved ROI are essential in being able to justify embarking on a release practice. Example ROI calculations have been provided.

- The business customer can be the most critical and most demanding of any group connected with the release process. The criticality comes from the high expectation of delivery of a quality service that will improve efficiency and in turn increase revenue. Creating a strategy of how to justify release management to the business is essential to the success of the initiative. Without the business customer understanding the benefits and the WIIFM, it will be difficult to maintain an effective release schedule and gain all of the benefits from a release process. The business customer must become an active partner in the process.

© 2010 by Taylor and Francis Group, LLC

Appendix A: Release Policy*

Contents

Revisions 129
Record of Past Modifications 129
Document Owner 129
Abbreviations 130
A.1 Overview 130
A.1.1 Purpose 130
A.1.2 Scope 131
A.1.3 Mission of Release Management 131
A.1.4 Strategy 132
A.1.5 Release Planning 132
A.2 Release Management—Organizational Context 132
A.2.1 Guidance 132
A.2.2 Facilitator 133
A.2.3 Governance 133
A.2.4 Objectives 133
A.3 Types of Releases 133
A.3.1 Release Unit 133
A.3.2 Full Release 134
A.3.3 Delta Release 134
A.3.4 Package Release 134
A.3.5 Release Categories 134

* Release Management
Release Management Policy
Version 1.0

© 2010 by Taylor and Francis Group, LLC

A.4 Release Naming Conventions 135
A.4.1 Naming Schema 135
A.4.2 Standard Naming Conventions 135
A.5 Release Schedule 136
A.5.1 Strategy 136
A.5.2 Scheduling 136
A.5.2.1 Example of a Release Schedule 137
A.5.3 Scheduling Restrictions 137
A.6 Definitive Media Library (DML) 137
A.7 Release Lifecycle 139
A.7.1 RLC Overview 139
A.7.2 Components of the RLC 139
A.7.3 Release Planning 140
A.7.3.1 Initiation 140
A.7.4 Design, Develop, Build, and Configure 140
A.7.4.1 New Environment Request Process (NER) 140
A.7.4.2 Application Installation 141
A.7.5 Release Acceptance 141
A.7.5.1 Quality Assurance (QA) 141
A.7.5.2 Operational Readiness (OR) 141
A.7.6 Communication, Preparation, and Training Plan 142
A.7.7 Installation and Distribution 142
A.7.7.1 Implementation 142
A.7.7.2 Post-Implementation 142
A.7.8 Release Documentation Retention 142
A.8 Roles and Responsibilities 143
A.9 Process Relationships 145
A.9.1 Incident Management/Service Desk 145
A.9.2 Problem Management 146
A.9.3 Change Management 146
A.9.4 Configuration Management 146
A.9.5 Service Level Management 146
A.9.6 IT Security Management 147
A.9.7 IT Service Continuity Management 147
A.9.8 Software from Developers and Suppliers 147
A.10 Key Performance Indicators (KPIs) 147
A.10.1 Critical Success Factors 147
A.10.2 Key Performance Indicators for Releases 148
A.10.3 Key Performance Indicators for Process Performance 148

© 2010 by Taylor and Francis Group, LLC

A.11 Management Reporting.....149
A.11.1 Meeting and Control Structures.....150
A.11.1.1 Change Advisory Board (CAB).....150
A.11.1.2 Release Review.....150
A.11.1.3 Postrelease Review.....150
Attachment A: Key Performance Indicators (KPIs).....150

Revisions

Record of Past Modifications

Below is a record of all modifications previously made to this document.

Version	*Date*	*Name*	*Details*

Document Owner

The individual named below is the owner of this document. No modifications can be made to the document without going through this individual. All modifications made to the document will be recorded by the document owner in the "Record of Past Modifications" section above.

<Document Owner's Name>	
Job Title:	
Company:	
Dept:	
Phone:	
Location:	
E-mail:	

© 2010 by Taylor and Francis Group, LLC

Abbreviations

Acronym	*Description*
CAB	Change Advisory Board
CI	configuration item
CMDB	configuration management database
COE	Center of Excellence
DML	definitive media library
CSF	critical success factor
ITIL	Information Technology Infrastructure Library
KEDB	known error database
KPI	key performance indicator
NER	New Environment Request (phase of the RLC)
OR	Operational Readiness
PM	project manager
PMO	project management office
RFC	requests for change
QA	Quality Assurance (phase of the RLC)
RLC	release lifecycle
RACI	responsible, accountable, consulted, and informed
RM	release management
SDLC	software development lifecycle
SLM	service-level management
SM	Service Management (short for ITIL Service Management)
UAT	user acceptance testing

A.1 Overview

A.1.1 Purpose

The release policy serves as a guide to the overall release strategy for the organization. It has been developed to support the organization's Mission Statement to address the needs of business customers and users. On a high level, it defines the roles and responsibilities of release management within the application and system development process and clarifies how the release lifecycle (RLC) helps to ensure production-ready implementations. The release policy will also define release standards that should be utilized.

- Types of releases and frequency
- Release naming conventions

© 2010 by Taylor and Francis Group, LLC

- Outlines release time restrictions
- Critical project deliverables
- Interrelationship with other Information Technology Infrastructure Library (ITIL) processes, the configuration management database (CMDB), and the definitive media library (DML)

A.1.2 Scope

Release management is the recognized process that will be utilized for the delivery of all application and system releases. Key control artifacts identified within the release process are used to govern the release process. These control artifacts align with those identified with the agreed-upon release lifecycle. All applications under the control of release management, including vendor-supplied software, are subject to the release policy.

Software developed and delivered by vendors is also subject to the release management process. Vendors are responsible for supplying all appropriate process support and documentation deliverables.

All deliverables are to be managed from development or purchasing, to customization and configuration, testing, and implementation through to operation in the production environment.

A.1.3 Mission of Release Management

Release management undertakes the planning, design, building, configuration, and testing of software and hardware to create a set of release components for the live environment. Activities also cover the planning, preparation, and scheduling of a release to many customers and locations.

The mission of release management (RM) is to establish a process that achieves a level of effectiveness that enables the organization to develop and deliver products in a manner that is:

- Repeatable—enables measurement of the process
- Controllable—measures how well we are doing and where improvements can be made
- Scalable—enables support of all future projects

This leads to:

- Consistency of release quality
- Reduction in of the number of incidents
- Greater efficiency in planning of and coordination of changes to the organization's environment

In doing so, release management will enable the organization to be able to cope with a high frequency of software and hardware releases without sacrificing IT

© 2010 by Taylor and Francis Group, LLC

service quality. The control mechanisms within release management help support this objective in an efficient and economical manner.

The benefits are:

- Delivery of quality releases that are scalable for the future
- Documented policies and procedures that ensure that all stakeholders understand release and change requirements
- Efficient, consistent, and repeatable processes

A.1.4 Strategy

Release management plays an integral role in ensuring the integrity of all application and system development by following a consistent and repeatable process. There is a close linkage between release management and the project management office (PMO). Release management, however, is not the development lifecycle. The PMO provides guidelines for the development lifecycle within project management. The role of release management is to assist project management with the planning and implementation of the release into the production environment. The PMO is responsible for tracking and reporting on the status of projects.

Release management enables the development lifecycle. It does so by providing the process structure and defining the related procedures. It also provides artifact templates that enable the project team to efficiently carry out the required development tasks associated with release deliverables. The release lifecycle (RLC) defines the artifacts that provide the basis for project governance.

A.1.5 Release Planning

The planning process ensures that necessary resources are available, thorough testing has been done, and that turnover documentation has been completed in order to ensure a smooth implementation. Release management plays a key role, along with project management, in the resource planning process. Utilization of the release lifecycle (RLC) guides the release planning process. The interrelationships among release, change, and configuration management are essential to the effective population and use of the CMDB. Effective release planning ensures accurate population of the CMDB, which has an effect on other service support and delivery processes.

A.2 Release Management—Organizational Context

A.2.1 Guidance

One of the primary activities of release management is to define and manage the RLC, which provides guidance throughout the development of releases. Defining the RLC and consulting with design teams through the process ensures the

© 2010 by Taylor and Francis Group, LLC

successful releases of applications and associated hardware. Release management works closely with the PMO to provide training to the project manager's processes and the key deliverables throughout the RLC.

A.2.2 Facilitator

Throughout the RLC there are multiple process checkpoints, phase approvals, and review meetings. Release management will assist the project manager (PM) in managing and tracking these milestones. Additionally, release management may be aware of projects or releases that conflict with each other. In these cases, release management would facilitate a review session with the project teams to assist in the resolution of any issues arising from scheduling conflicts.

A.2.3 Governance

While the primary role of release management is not governance, it does play a joint governance role with the PMO. The primary role is to ensure that predefined RLC documentation is utilized and completed as required. The RLC documentation provides the templates and defines the procedures to ensure that proper testing, scheduling, and operation readiness is completed prior to release implementation.

A.2.4 Objectives

- Establish a process that will raise the effectiveness of the release process to maintain a consistent level of quality for system and software delivery.
- Minimize service disruptions to the business by ensuring efficient coordination of changes to the infrastructure through release planning.
- Enable better support for new applications by defining the required deliverables and acceptance criteria for each type of release and maintaining those artifacts in a secure definitive media library (DML).
- Monitor the implementation of new software and hardware releases into the environment and integrate with all related service management processes.
- Ensure that the release management process is fully integrated across all development groups and clearly defined roles and responsibilities exist for each group involved in the process.

A.3 Types of Releases

A.3.1 Release Unit

A release unit is the segment of software or hardware that is normally released together. These units may vary depending on the type and level of configuration items (CI) within the infrastructure. Release units can be defined differently for systems or

© 2010 by Taylor and Francis Group, LLC

infrastructures due to the differences in the interrelationships. The purpose is to define the most appropriate release unit level for each application or type of application.

When deciding the appropriate level for release units, the development team should consider the following:

- The amount of change necessary at each level
- The amount of resources and time required
- The ease of implementation
- The complexity of interfaces with the infrastructure
- The available capacity of the different environments

A.3.2 Full Release

All CIs of a release unit are developed, tested, and implemented together. Since a full release considers all CIs within a release unit, there is less chance of having conflicts since all CIs and interfaces will be tested, even if unchanged during the development of the release.

A.3.3 Delta Release

A delta release is also known as a partial release or one that only includes one or many but not all of the configuration items (CIs) within a release unit. A delta release can have a greater risk than a full release due to lack of regression testing and linkage to all CIs within the release unit. In some instances, a delta release is more appropriate than a full release due to time, urgency, or cost justification. Careful evaluation of unchanged or older CIs/interfaces must be done when considering a delta release.

A.3.4 Package Release

Package releases are utilized when there is linkage between a release unit and other systems, interfaces, or release units. A package release can consist of both a full release and several delta releases. Where appropriate, a package release can reduce the risk of outdated or incompatible software being used. It can also ensure testing for compatibility between systems and interfaces.

A.3.5 Release Categories

Major software releases and hardware upgrades—Large amounts of new functionality. Usually a major release supersedes all preceding minor upgrades, releases, and emergency fixes. These types of releases can include architecture changes, partial redesigns, and rewrites.

Minor software releases and hardware upgrades—Consists of small enhancements, functional changes, problem fixes, and technical updates.

© 2010 by Taylor and Francis Group, LLC

Emergency software and hardware fixes—Small enhancements or fixes usually implemented in response to an incident or known error. Includes patch management and problem fixes that cannot be postponed or custom requirements that cannot be postponed (e.g., regulatory changes).

A.4 Release Naming Conventions

A.4.1 Naming Schema

A naming schema uniquely identifies the CI it represent and includes a version numbering system that follows a universal nomenclature that is consistent for all releases and across all functional areas.

A.4.2 Standard Naming Conventions

All releases will follow a standard naming convention for all components released into the infrastructure to ensure consistency and easy identification as to the type of release and to ensure version control. Release naming conventions will include the CI name, major release number, minor release number, and emergency release number.

Examples of this convention are: [application name] X.Y.Z; for example, CIMS 1.0.0

Category	*Version Number That Changes*	*Proposed Release Unit*
Major Release	X	Full
Minor Release	Y	Full
Emergency	Z	Delta

- Major Release: CRM v1.0, v2.0, etc.
- Minor Release: e-mail v1.1, v1.2, etc.
- Emergency Fix Release: CRM v1.1.1, v1.1.2, etc.

When a new version of an application is developed, the version number should be immediately updated for the software in the development area.

Example: When CRM v1.0.0 is operational and a new version of a CRM component is to be developed (for example, a minor release), a copy of CRM v1.0.0 is taken from the DML, moved to the development environment, and labeled as CRM v1.1.0a.

Every time a functional change is made (minor release), the number is increased by one letter, so the next version is 1.1.0b. Once the software is ready to be deployed, it is named v1.1.0 and the master copy is stored in the DML.

© 2010 by Taylor and Francis Group, LLC

It is the responsibility of the development team to determine what type of release is being developed; this will determine the naming convention.

A.5 Release Schedule

A.5.1 Strategy

A release can be viewed as many changes to a release unit, a service, or a CI. The organization of these changes into a regular release schedule allows for a more controlled and quality implementation. A regular release schedule should be established for each application that will allow adequate time to plan throughout the entire RLC. The type of release and timing should be based on the complexity and urgency of the changes to the release unit. A monthly release schedule for minor releases will allow for the gathering of changes as well as a controlled method of release. Additionally, a predetermined release schedule will set the expectation of when users can anticipate updates to their systems.

A.5.2 Scheduling

A preset release strategy must be established to ensure proper planning and testing. A standard release schedule must be designated for each application and must be adhered to by all functional teams. Each release must be designated by the standardized release numbering convention mentioned in Section A.4.2.

It is the responsibility of each application group to determine the number and frequency for each release type. This release schedule will then become part of the release policy and govern the releases for that application. This information will be provided to change management and used to evaluate and grant authorization for requests for change (RFC).

For new applications, establishment of the release schedule must be done during the initial design and development of the system and documented in the operations readiness document. When establishing the release schedule several considerations must be reviewed.

- How often will minor changes need to be made to the application?
- How long will it take to develop and test changes?
- Does the standard release schedule allow time for evaluation of the last release?
- How often should a complete review of the application be done and what would warrant a major change?

Existing applications must also create a release schedule and move to the release methodology rather than making minor changes as they arise. This allows for better historical tracking and documentation of changes.

© 2010 by Taylor and Francis Group, LLC

A.5.2.1 Example of a Release Schedule

A particular application is released eight times per year during the scheduled release period.

- Eight times a year; this indicates a maximum of two major releases and six minor releases.
- Emergency releases on an as-required basis.

Type of Release/ Numbering	*Schedule*	*Issues Addressed*
Minor v1.**1**.0, 1.**2**.0 ….	6 to 8 weeks	■ Minor changes to functionality ■ Security changes ■ Changes to parts of the release unit
Major v**1**.0, **2**.0 …	Every 6 to 12 months	■ Major revision to functionality ■ Complete change in look and feel ■ Change to entire release unit
Emergency v1.1.**1,** v1.1.**2** …	As needed	■ Changes that must be made that are outside of the established release schedule

A.5.3 Scheduling Restrictions

When planning a release schedule, consideration must be given to business-critical times to avoid, as well as other restrictions within the environment, such as month-end freezes. The change management process plays a critical role in the scheduling of releases to ensure these times are avoided.

Release management and project management calendars should be synchronized so as to reflect the same freeze period and release dates.

A.6 Definitive Media Library (DML)

The definitive media library (DML) is the term used to describe a secure location in which the definitive authorized versions of all software CIs are stored and protected. This storage area may in reality consist of one or more software libraries or file storage areas that should be separate from the development, test, and production environments. It contains the master copies of all controlled software in an organization.

© 2010 by Taylor and Francis Group, LLC

The DML should include definitive copies of purchased software (along with license documentation or related information), as well as software developed internally. Master copies of controlled system documentation will also be stored in the DML in electronic form.

All releases will be controlled through software configuration management and protected in the DML throughout their lifecycles.

Title	*Lockdown*
Policy Statement	■ All releases should be controlled through the software configuration management procedure and protected in the DML throughout their lifecycles. ■ Source code can only be approached through tools that protect the integrity of the sources. ■ Direct approach of repositories is not allowed. ■ All source code is labeled and versioned.
Purpose	■ To protect the software assets ■ To secure the assets and the IP ■ To ensure the integrity of the source code by limiting access to the repositories ■ To prevent business continuity issues related to software development
Scope of Policy	■ All of IT
Benefits	**A lockdown will:** ■ Improve quality checks on all software developed by clear check-in and checkout procedures ■ Limit unauthorized access to zero, therefore increasing integrity ■ Improve business continuity capabilities through centralized backups ■ Allow for better tracking of code migrations through their lifecycle, knowing who is touching what asset at what time and why
Related Procedures and Models	**Tools:** ■ <list any tools if applicable>
Effective Date	TBD

© 2010 by Taylor and Francis Group, LLC

A.7 Release Lifecycle

A.7.1 RLC Overview

Planning, testing, and documenting these activities are the main objectives of release management. These objectives can be achieved by following a rigorous, consistent, and repeatable process that is not obtrusive. This helps to ensure that proper requirements are gathered, that the application is tested, is operationally ready for production, and has a smooth implementation into the production environment.

The RLC is not a system/software development lifecycle (SDLC); rather, the RLC complements the SDLC and guides the release of the software/system by ensuring that all releases have been planned, tested, and scheduled.

A.7.2 Components of the RLC

The RLC is aligned with the development lifecycle. The PMO plays an active role in the RLC from an integration standpoint, through project managers and the development teams.

The RLC consists of seven different phases; each phase relates to the other phases that lead to a production-ready release. The RLC phases are:

- Initiation
- New Environment Request (NER)
- Release configuration and build
- Quality Assurance (QA)
- Operations Readiness (OR)
- Implementation
- Post-implementation

Each phase requires the creation of specific deliverables that will help to ensure that each release is properly designed, constructed, tested, and operationally ready prior to implementation. Each phase concludes with a final review that must be passed prior to moving to the next phase. This review coincides with the phase outlined in the development lifecycle.

Additionally, the RLC is based on the seven major activities defined in the ITIL release management framework. The RLC adapts the theoretical concepts of ITIL and incorporates the practical application of those concepts into the environment. Each of these phases are related to the other phases, and together, lead to a production-ready release.

The ITIL release management activities are:

- Planning
- Development and design

© 2010 by Taylor and Francis Group, LLC

- Build and configure
- Release acceptance
- Rollout planning
- Communication, preparation, and training
- Distribution and installation

A.7.3 Release Planning

Release management plays an integral role in ensuring the integrity of all application and systems development by following a consistent and repeatable planning process. The planning process ensures that needed resources are available, thorough testing has been completed, and turnover documentation has been completed.

A high-level plan is required for every release and should include all test plans, acceptance criteria, and release deliverables.

A.7.3.1 Initiation

Initiation is the entrance phase into the RLC and begins after the project has been approved. Several documents, such as a concept proposal, business case, feasibility analysis, project charter, and project plan and business requirements will be completed prior to entering this phase. An initiation checklist is used to enter this phase. The main purpose of the Initiation phase is to ensure that project and budget approval have been given and that business requirements and deliverables have been clearly defined.

A.7.4 Design, Develop, Build, and Configure

A.7.4.1 New Environment Request Process (NER)

The new environment request process ensures that the architecture of the release is configured to provide the most efficient use of hardware, network, storage, and database resources. It defines a mechanism for determining whether existing infrastructure can be utilized or if procurement of new components is necessary. The process defines a procedure for navigating the procurement process. Additional process outputs include data flow, process flow, and architecture overview diagrams, which can be used in the future support of the application.

Releases not within the NER process are also under the control of the change and release management processes.

During the NER process, the software development lifecycle (SDLC) is ongoing in parallel with the NER process.

© 2010 by Taylor and Francis Group, LLC

A.7.4.2 Application Installation

Application Installation is the procedure that describes the interface between the software development lifecycle and release management. It is the responsibility of release management to ensure that all trusted source code and release components are secured in the definitive media library (DML) prior to deployment. When the environment is built according to the requirements and has been validated, the application can be safely installed.

The NER process is ongoing in parallel with Application Installation.

A.7.5 Release Acceptance

The procedures for Release Acceptance and Testing are covered under the QA and OR processes.

A.7.5.1 Quality Assurance (QA)

In order to provide the best production-ready product, the release needs to go through a defined QA process. The test cases performed need to map directly back to the business requirements. The QA process should also include performance testing as well as user acceptence testing (UAT). The testing process may identify bugs within the release; the project team must review these bugs and determine fixes or workarounds. The RLC requires that the QA Acceptance Report be reviewed to ensure the QA group has either endorsed the release or has properly identified potential bugs that may cause issues in production. The report also ensures that the project team has acknowledged the associated risk.

A.7.5.2 Operational Readiness (OR)

Operational Readiness (OR) is the final process prior to implementing the release into production. The focus of the process is to ensure that all components of the release have been tested, reviewed, and documented. This is achieved with the final operations readiness review. The review serves as a checklist of required artifacts and helps to ensure that the release is ready for promotion to the production environment. Documentation includes release support information, any known bugs and workarounds, service continuity measures, service level documentation, training and implementation plans, and technical operational readiness testing (ORT).

ORT is a rigorous process that is designed to test and ensure that the infrastructure, network, and databases perform according to business requirements. It also ensures that monitoring tools are in place and that all updates to services and CIs are included in the CMDB.

© 2010 by Taylor and Francis Group, LLC

A.7.6 Communication, Preparation, and Training Plan

The goal of the Communication Plan is to inform all stakeholders of all issues relevant to the release.

It is the responsibility of the project team to ensure that the deliverables listed below are included with the release. These artifacts are included as part of the RLC processes.

- Support and Escalation Plan/Turnover document (OR process)
- Master Training Plan (OR process)
- Implementation and Communication Plans (Implementation process)

A.7.7 Installation and Distribution

Procedures should be planned and documented for implementing the release; these standard procedures should be reused where possible. Proper planning of the installation and distribution of the release are crucial to the success of the release.

A.7.7.1 Implementation

Release management works hand in hand with change management to ensure that the steps required to ensure a successful implementation are completed prior to receiving approval to implement. These steps include communication, implementation, validation, criteria for success, and back-out plans. These steps are captured within the RFC. Once the RFC is completed and approved by the Change Advisory Board (CAB), the release can then move into production according to the implementation plan.

A.7.7.2 Post-Implementation

The post-implementation process is the final process within the RLC. The main objective of this process is project closure. During this phase, all project documents are gathered, reviewed, and archived. Additionally, a post-implementation review (PIR) meeting may be necessary to gather the lessons learned and gain knowledge to improve future releases.

A.7.8 Release Documentation Retention

RLC documentation generated during the development process is stored in the document repository. This allows for a central repository for this documentation.

© 2010 by Taylor and Francis Group, LLC

A.8 Roles and Responsibilities

An RACI (responsible, accountable, consulted, and informed) matrix is supplied to provide the roles and responsibilities for release management.

Activities	*Service Manager*	*Release Manager*	*Release Coordinator*	*Project Manager*
Overall design and ongoing maintenance of process, including the RLC, artifacts, and release policy	R	A	C	I
Conducting ongoing process assessment of opportunities for improvement	C	A/R	C	I
Ensuring people, process, and technology are aligned	A	R	I	
Establish key performance indicators (KPIs) that measure critical success factors	R	A	C	I
Holds a holistic view of the organization as a whole and ensures proper alignment between relevant service management processes as well as functional areas	A	C		
Act as a process champion and promote the process to senior manager	A	R		
Provides required training of the process	C	R	C	I
Ensuring key goal of RM is to use standard methods and artifacts in preparation, building, testing, and implementation of changes with minimal impact	R	A	C	

© 2010 by Taylor and Francis Group, LLC

Activities	*Service Manager*	*Release Manager*	*Release Coordinator*	*Project Manager*
Produce and monitor KPIs and critical success factors (CSFs) to ensure maximized efficiencies	C	A/R	C	I
Act as the focal point operationally for proposed modifications to release management activities	R	A/R		
Act as central point of contact for the process team		A/R		
Supports Release Manager role in carrying out activities	C	I	A/R	
Maintain and track information concerning releases throughout the RLC, including status, release artifacts, and closure		A/R	R	I
Produce reports by release type		A/R	R	
Produce and maintain appropriate communications to all stakeholders regarding releases	C	R	C	A
Coordinate activities throughout the RLC	C	R	R	A
Administrative support to the Release Manager and the process	C	C	A/R	
Ensure that the requirements for the specific RLC are planned and executed to meet release schedule		I	C	A/R
Completion of release artifacts as outlined in RLC		I	C	A/R

© 2010 by Taylor and Francis Group, LLC

Activities	*Service Manager*	*Release Manager*	*Release Coordinator*	*Project Manager*
Completion of release artifacts as outlined in RLC		I	C	A/R
Sign off on all RLC control points	I	R	C	A/R
Ensures that master copies of all software are secured in the DML	I	A	I	R
Ensures the successful release of all software and hardware	I	A	I	R
Plan the release and implementation	I	A	I	R
Execute the transfer of deliverables (support, training, and updated service information) to the Service Desk	I	A	I	R
Communicate, prepare, and train end users	I	A	I	R
Verify testing results and deliverables	I	A	I	A/R
Ensures close relationship exists between release management and all other Service Management (SM) processes	A/R	C		

Note: R – Responsible, A – Accountable, C – Consulted, I – Informed

A.9 Process Relationships

A.9.1 Incident Management/Service Desk

During the RLC, support documentation for the application is created. Information relating to password resets, escalation procedures, and support documentation is created and provided to the Service Desk and Incident Management. Additionally, during the testing phase of the RLC, known errors may be discovered and

© 2010 by Taylor and Francis Group, LLC

documented in the known error database (KEDB); if there is a workaround, this is also documented in the Service Desk knowledge base.

A.9.2 Problem Management

Documented known errors and associated workarounds will help to resolve incidents. This information can then be used to provide justification for identification and further study of the problem. Documentation of test results and operations readiness information can also be reviewed during the problem-resolution process and used for further root cause analysis and known error identification.

A.9.3 Change Management

Change management plays a key role in the release management process in helping to ensure that scheduling of the release is completed in an expeditious yet controlled manner. All phases and processes of the RLC lead to the RFC and the approval of the change. Change management also helps to ensure that changed CIs are documented in the CMDB.

Although release management oversees the details of the implementation of a change, it remains under the control and authority of change management.

A.9.4 Configuration Management

Before a release can be developed, there should be an understanding of the existing application, system, the associated CIs, and the relationship to other release units. This information is stored in the CMDB and managed by configuration management. As the new application is developed, configuration of the application is captured and documented through the RLC process. This information is subsequently used to populate the CMDB.

In partnership with service level management, the service catalog is also maintained within the CMDB (as a CI) and should be updated as the service or application changes.

A.9.5 Service Level Management

In the development of the release, system availability and performance requirements, support service levels, and maintenance windows are determined by the customer. These considerations should be understood so the proper sizing and architecture of the application can be designed to meet the desired levels. Service level management maintains the baseline availability and service levels as well as the schedules and costs associated with nonbaseline variables. Service level management will also work with configuration management to ensure that the service catalog is updated to reflect changes to services or service levels.

© 2010 by Taylor and Francis Group, LLC

A.9.6 IT Security Management

One of the activities required during RLC is to ensure the security of the enterprise by thoroughly understanding whether there are any related risks or vulnerabilities to the application or the connectivity. In the RLC, a security assessment and audit is required to ensure compliance with corporate security policies and to determine potential vulnerabilities.

A.9.7 IT Service Continuity Management

Continuation and restoration of service is documented within the RLC. This documentation becomes part of the overall IT enterprise Service Continuity Plan.

A.9.8 Software from Developers and Suppliers

When software is accepted from developers or suppliers, it is placed in the DML and registered in the CMDB. Release management is responsible for ensuring that the new releases are built only from the DML into the controlled test environment. When testing has been successfully completed for all environments, release management will oversee the implementation of the release into the production environment.

A.10 Key Performance Indicators (KPIs)

In order to monitor the release management process, a number of key performance indicators (KPIs) are monitored to assess the effectiveness of the release management process. The most important KPIs to focus on are presented below:

The four critical success factors (CSFs) for release management are:

- Better quality software and hardware
- A repeatable process for rolling out software and hardware releases
- Implementation of releases swiftly (business-driven needs) and accurately
- Cost-effective releases

CSFs relate to key performance indicators as follows.

A.10.1 Critical Success Factors

The success of release management can be determined as follows:

Repeatability—How many applications follow the release process?
Controllability—How many exception releases (and incidents) do we have outside the release cycle that are caused by bad product?

© 2010 by Taylor and Francis Group, LLC

Scalability—How well are we able to incorporate products from external vendors in our release cycle so that they are implemented successfully at release day?

A.10.2 Key Performance Indicators for Releases

The key performance indicators (KPIs) associated with the critical success factors (CSFs) are:

CSF	*KPIs*
Repeatability	■ % applications following the release process > X% (to be decided) ■ % reduction in build failure
Controllability	■ # of exception releases for in-scope applications is down by Y% ■ # of times code is moved directly into QA or production without testing ■ # of applications that have a completely defined major and minor release set and their release units ■ % of applications stored in the DML
Scalability	■ % of changes by outsourced vendors that are managed through release management ■ % of outsourced vendor projects that are in sync with the release calendar ■ % of COTS (commercial, off-the-shelf) software that is managed through release management

A.10.3 Key Performance Indicators for Process Performance

Key performance indicators (KPIs) allow measurement of how well the process performs to achieve the critical success factors (CSFs). KPIs do not measure results necessarily, but give valuable insight into how the process comes to its results.

CSF	*KPIs*
Better quality software and hardware:	■ % reduction in the use of software and hardware releases that have not passed the required quality checks ■ % of installed software not taken from the DML

© 2010 by Taylor and Francis Group, LLC

CSF	*KPIs*
Repeatable process for rolling out software and hardware releases:	■ # of new releases planned and controlled by release management ■ % reduction of urgent releases (exception and emergency releases) ■ Reduction in the number of urgent releases ■ % increase of normal release units as opposed to ad hoc releases
Implementation of releases swiftly (business-driven needs) and accurately:	■ % reduction of releases implemented without being tested ■ % reduction in the releases causing incidents
Cost-effective releases:	■ % reduction of failed releases ■ # of times that there was a reduction in the service unavailability caused by releases ■ Increased % of releases built and implemented on schedule ■ % reduction in release overtime due to better planning ■ % increase in accuracy of release estimates

Complete list of KPIs considered can be found in Attachment A at the end of this document.

A.11 Management Reporting

Management reports serve to monitor, check, and improve the outcomes of the process, and to share information with other processes. The release management reports serve to measure and monitor the quality of the process against the metrics established as KPIs to ensure that it is efficient, effective, and fit for its purpose.

Two types of release reports will be established:

- Process reports
- Release reports (per application)

© 2010 by Taylor and Francis Group, LLC

The process reports will be based on the CSFs for release management and will reflect the cumulative performance over the year. Frequency is per release.

The release reports per application will be created on a weekly basis and feed into the Process Report, which will be distributed to management.

A.11.1 Meeting and Control Structures

A.11.1.1 Change Advisory Board (CAB)

The CAB authorizes the release and its contents. It reviews the planning preparation, the release contents, and the impact analysis. It decides to move forward by granting approval for implementation.

A.11.1.2 Release Review

This review examines the final preparation before the implementation of a release.

All required release artifacts are evaluated for completion. The results will be reported to the change manager, who will finally authorize the release for implementation.

A.11.1.3 Postrelease Review

This review is designed to address all issues and tie up loose ends in production, and to improve the preparation for future releases.

The output will include a review document with action items and improvement plans. This review document will be sent to change management to be included in the post-implementation Review for change management.

Attachment A: Key Performance Indicators (KPIs)

1. Better quality software and hardware:
 a. Percentage reduction in the use of software and hardware releases that have not passed the required quality checks
 b. Percentage reduction in installed software not taken from the DML
 c. Percentage reduction of unauthorized reversion to previous releases
2. Repeatable process for rolling out software and hardware releases:
 a. All new releases planned and controlled by release management
 b. All installed software taken from the DML
 c. Percentage reduction in the number of failed distributions of releases to remote sites
 d. Reduction in the percentage of urgent releases
 e. Increase in the percentage of normal release units as opposed to ad hoc releases

© 2010 by Taylor and Francis Group, LLC

3. Implementation of releases swiftly (business-driven needs) and accurately:
 a. Percentage reduction in build failures
 b. Percentage of implementation of releases at all sites, including remote ones, on time
 c. Percentage reduction in the number of urgent releases
 d. Percentage reduction in the releases causing incidents
 e. Reduction in the percentage of releases implemented without being tested
 f. Reduced percentage of urgent or high-priority releases requested without the appropriate business case justification
4. Cost-effective releases
 a. Increased percentage of releases built and implemented on schedule
 b. Percentage releases built and implemented within budget
 c. Reduction in the service unavailability caused by releases
 d. Percentage reduction in releases backed out
 e. Percentage reduction of failed releases
 f. Percentage reduction in release overtime due to better planning
 g. Reduction in the "cost" of failed releases
 h. Percentage improvement of the planned composition of releases matching the actual composition (which demonstrates good release planning)

© 2010 by Taylor and Francis Group, LLC

Appendix B: RACI Matrix

Activities	*Service Manager*	*Release Manager*	*Release Coordinator*	*Release Project Manager*	*Release Control Board*
Overall design and ongoing maintenance of process, including the release lifecycle (RLC), artifacts, and release policy	R	A	C	I	
Conducting ongoing process assessment of opportunities for improvement	C	A/R	C	I	C
Ensuring people, process, and technology are aligned	A	R	I		C
Establish key performance indicators (KPIs) that measure critical success factors	R	A	C	I	C

© 2010 by Taylor and Francis Group, LLC

Activities	*Service Manager*	*Release Manager*	*Release Coordinator*	*Release Project Manager*	*Release Control Board*
Holds a holistic view of the organization as a whole and ensures proper alignment between relevant service management processes as well as functional areas	A	C			
Act as a process champion and promote the process to senior manager	A	R			I
Provides required training of the process	C	R	C	I	
Ensuring key goal of release management (RM) is to use standard methods and artifacts in preparation, building, testing, and implementation of changes with minimal impact	R	A	C		
Produce and monitor KPIs and critical success factors (CSFs) to ensure maximized efficiencies	C	A/R	C	I	I

© 2010 by Taylor and Francis Group, LLC

Activities	*Service Manager*	*Release Manager*	*Release Coordinator*	*Release Project Manager*	*Release Control Board*
Act as the focal point operationally for proposed modifications to release management activities	R	A/R			C
Act as central point of contact for the process team		A/R			
Supports release manager role in carrying out activities	C	I	A/R		C
Maintain and track information concerning releases throughout the RLC, including status, release artifacts, and closure		A/R	R	I	I
Produce reports by release type		A/R	R		
Produce and maintain appropriate communications to all stakeholders regarding releases	C	R	C	A	
Coordinate activities throughout the RLC	C	R	R	A	

© 2010 by Taylor and Francis Group, LLC

Activities	*Service Manager*	*Release Manager*	*Release Coordinator*	*Release Project Manager*	*Release Control Board*
Administrative support to the release manager and the process	C	C	A/R		
Ensure that the requirements for the specific RLC are planned and executed to meet the release schedule		I	C	A/R	
Completion of release artifacts as outlined in RLC		I	C	A/R	I
Sign off on all RLC control points	I	R	C	A/R	R
Ensures that master copies of all software are secured in the definitive media library (DML)	I	A	I	R	
Ensures the successful release of all software and hardware	I	A	I	R	I
Plan the release and implementation	I	A	I	R	I
Execute the transfer of deliverables (support, training, and updated service information) to the Service Desk	I	A	I	R	

© 2010 by Taylor and Francis Group, LLC

Activities	*Service Manager*	*Release Manager*	*Release Coordinator*	*Release Project Manager*	*Release Control Board*
Communicate, prepare, and train end users	I	A	I	R	
Verify testing results and deliverables	I	A	I	A/R	C
Ensures close relationship exists between release management and all other Service Management (SM) processes	A/R	C			

Note: R – responsible, A – Accountable, C – Consulted, I – Informed

© 2010 by Taylor and Francis Group, LLC

Appendix C: Release Deliverables Checklist

Phase Deliverable	*Release Management (RM) Checklist*	*Notes*
Initiation		
Concept Proposal/Internal Concept Proposal	☐	
Estimation Requirements	☐	
Business Requirements	☐	
Project Charter	☐	
Release Deliverables Checklist	☐	
Release Project Plan	☐	
Release Configuration and Build		
Functional Requirements	☐	
Master Test Plan	☐	
Service Level Agreement (SLA)	☐	
Performance Test Questionnaire	☐	

© 2010 by Taylor and Francis Group, LLC

Phase Deliverable	*Release Management (RM) Checklist*	*Notes*
New Environment Request (NER)		
New Environment Request Form	☐	
Application Architecture Design	☐	
Infrastructural Detail Design	☐	
Functional Specifications	☐	
Build Procedure	☐	
Performance Test Plan	☐	
Test Execution Plan	☐	
Quality Assurance/Testing		
Build Notes	☐	
Quality Assurance (QA) Progress Report	☐	
Quality Assurance (QA) Acceptance	☐	
Operational Readiness		
Operation Readiness Testing (ORT) Manual/Key Application	☐	
Support & Escalation/Service Desk Turnover	☐	
Master Training Plan	☐	
IT Service Continuity Plan	☐	
Implementation		
Implementation and Back-out Plan	☐	

© 2010 by Taylor and Francis Group, LLC

Phase Deliverable	*Release Management (RM) Checklist*	*Notes*
Project Implementation Notification	☐	
Release Notes	☐	
Change Control (FRC/Online Form, Change Advisory Board [CAB], Schedule)	☐	
Post-Implementation		
Post-implementation Review and Project Deliverables Signoff	☐	
Lessons Learned	☐	

© 2010 by Taylor and Francis Group, LLC

Appendix D: Master Test Plan*

Contents

Document Review 164
Reviewers 164
Stakeholders 165
Revisions 165
Record of Past Modifications 165
Document Owner 166
Acronyms and Terms 166
D.1 Introduction 170
D.1.1 Project Overview 170
D.1.2 Scope of This Document 170
D.1.3 Reference Documents 170
D.2 Testing Scope 171
D.2.1 Scope Inclusions 171
D.2.2 Scope Exclusions 171
D.3 Roles and Responsibilities 171
D.4 High-Level Test Plan 172
D.4.1 Unit Level 172
D.4.1.1 Unit Testing 172

* [Name of project]
Master Test Plan
Version x.x

© 2010 by Taylor and Francis Group, LLC

D.4.2 Integration Level Testing 174
D.4.2.1 Integration Testing 174
D.4.3 System Level Testing 175
D.4.3.1 Smoke Testing 175
D.4.3.2 System Testing 175
D.4.3.3 Regression Testing 176
D.4.3.4 Security Testing 176
D.4.3.5 Systems Interface Testing 177
D.4.3.6 Conversion Testing 177
D.4.3.7 End-To-End Testing 178
D.4.3.8 Failover Testing 178
D.4.3.9 Performance Testing 179
D.4.4 Acceptance Level Testing 179
D.4.4.1 User Acceptance Testing 180
D.4.4.2 Operational Readiness Testing 180
D.5 Schedule 181
D.6 Defect Management and Resolution 181
D.7 Communication 181

Document Review

Signing below indicates that you have been given the opportunity to review and provide feedback on the contents of this document.

Reviewers

Reviewers may be added or removed as needed.

_____ _____ Date
<Area> – <Reviewer Name>

_____ _____ Date
<Area> – <Reviewer Name>

_____ _____ Date
<Area> – <Reviewer Name>

_____ _____ Date
Quality Assurance – <Reviewer Name>

_____ _____ Date
Architect – <Reviewer Name>

© 2010 by Taylor and Francis Group, LLC

_____ _____ Date
Net Ops – <Reviewer Name>

_____ _____ Date
Tech Ops – <Reviewer Name>

_____ _____ Date
Storage – <Reviewer Name>

_____ _____ Date
Functional Team – <Reviewer Name>

Stakeholders

Additional stakeholders may be added as needed, but those listed below may not be removed.

Signing below indicates that you have been given the opportunity to review and provide feedback on the contents of this document.

_____ _____
<Business Area> – <Reviewer Name>/<Title> Date

_____ _____
Project Manager/Release Project Manager – <Reviewer Name>/<Title> Date

_____ _____
Architecture – <Reviewer Name>/<Title> Date

_____ _____
Operations – <Reviewer Name>/<Title> Date

_____ _____
IT Security – <Reviewer Name>/<Title> Date

_____ _____
Quality Assurance – <Reviewer Name>/<Title> Date

_____ _____
Release Management – <Reviewer Name>/<Title> Date

Revisions

Record of Past Modifications

Below is a record of all modifications previously made to this document.

© 2010 by Taylor and Francis Group, LLC

Version	*Date*	*Name*	*Details*

Document Owner

The individual named below is the owner of this document. No modifications can be made to the document without going through this individual. All modifications made to the document will be recorded by the document owner in the "Record of Past Modifications" section above.

[Name of document owner]	
Job Title:	
Company:	
Dept:	
Phone:	
Mail Drop:	
Location:	
E-mail:	

Acronyms and Terms

Acronyms and Terms	*Description*
bug	See defect.
conversion testing	Confirms the accuracy of the conversion procedures needed to initially load the data into the system. Also validates the usage of the data during day-to-day production activity. Performed during the system test level.

© 2010 by Taylor and Francis Group, LLC

Acronyms and Terms	*Description*
defect	The deviation of an actual result from the expected result during the application testing. A flaw in the software with potential to cause a failure that is raised by a tester and is meant for a developer to fix. If the defect cannot be resolved only by developers, then the item would be considered an issue.
end-to-end testing	Additional interface testing from the beginning to the end of a process including all upstream and downstream impacted systems that receive data, whether directly or indirectly from the primary system. Performed during the system test level.
entry criteria	Metrics specifying the condition that must be met in order to begin testing at the next stage or level.
environment	The collection of hardware, software, data, and personnel that comprise a level of testing.
exit criteria	Metrics specifying the conditions that must be met in order to promote a software product to the next stage or level.
failover testing	Tests the system availability and recovery processes in case of breakdown. Confirms that the load balancers, clustering on various pieces of the architecture, are working correctly. Performed during the system test level.
functional testing	See *system testing*.
integration testing	The objective of integration testing is to test the interaction of related data interface components in order to confirm that these components function properly when integrated together. This serves to identify and resolve major interface defects before starting system testing. Typically conducted by the development team.
interface testing	See *systems interface testing*.
interfacing system	Downstream or upstream system that may require change due to the primary system.

© 2010 by Taylor and Francis Group, LLC

Acronyms and Terms	*Description*
issue	An issue can be any discrepancies, deficiencies, or other abnormal system behavior noted during the entire phase of the software development lifecycle. Issues are not limited to hardware, database, network, and integration issues alone. Issues are often process related as well. An issue is anything that hampers the normal workflow during the system/software development lifecycle (SDLC). Items are considered to be issues if they require more than a developer to resolve. An issue can be raised by anyone related with the concerned application during any phase of the SLDC.
operational readiness testing (ORT)	ORT is performed to conclude that the application or new release can be deployed within the enterprise and supported by operations personnel. Performed during the acceptance test level.
performance monitoring	Proactive monitoring of application availability and performance in real time.
performance testing	Confirms that the system meets its end-user response time requirements and can handle increasing business volume. Load, high availability, production stability, growth, and stress are all performance tests. Performed during the acceptance test level.
primary system	Core application being developed or modified.
QA	Quality Assurance.
RACI	Responsible, accountable, consulted, and informed.
regression testing	Ensures that the changes made in the application have not introduced any new defects into the system. Performed during the system test level.
security testing	Confirmation that the system meets the project security requirements. Performed during the system test level.

© 2010 by Taylor and Francis Group, LLC

Acronyms and Terms	*Description*
smoke testing	A test run to demonstrate that the basic functionality of a system exists and that a certain level of stability has been achieved. Frequently used as a part of the entrance criteria for system testing. Performed during the system test level.
string test	See *integration testing*.
systems interface testing	Testing to see if data and control are passed correctly between directly interfacing systems (both real time and batch). Also called *interface testing*. Performed during the system test level.
system testing	A comprehensive test undertaken to validate the primary system and to ensure the business requirements are being met. Also called *functional testing*.
test data	Data that is entered into the test environment prior to the beginning of testing and is needed to successfully execute the test scripts. Data can come from production data or be mocked up data.
test level	Test levels are unit, integration, system, and acceptance.
unit testing	Focuses on testing the individual functional and technical components of the application before they are integrated with the rest of the system. Typically conducted by the developer who wrote the code.
user acceptance testing (UAT)	UAT ensures that the end users and stakeholders are satisfied with the application. UAT is not a detailed test, but allows end users to validate at a high level that the business requirements have been achieved. Performed during the acceptance test level.

© 2010 by Taylor and Francis Group, LLC

D.1 Introduction

[*The use of high-level diagrams is recommended to add clarity.*]

The purpose of a Master Test Plan is to orchestrate testing at all levels for a project. The goal of test planning is to address the testing strategy, testing scope, resource roles and responsibilities, schedule**,** and defect management. The Master Test Plan is a living document and will be versioned to accommodate changes in the project or testing approach.

D.1.1 Project Overview

[*State briefly the intent or the aim of the project. This section should be in accordance with the Project Charter or Project Plan.*]

D.1.2 Scope of This Document

[*State briefly what is covered in the document and what is not.*] Example: The purpose of this document is to define the overall test plan for project X.

This document covers the testing scope, testing approach, schedule, tasks, responsibilities, risks, assumptions, and dependencies associated with the testing. The document also contains general information regarding test cases. The list of test scripts (along with their testing objectives) that will be executed and the detailed steps of the test scripts are not covered in the scope of this document. They will be documented separately.

The detailed plans for test execution, performance testing, and failover testing will be covered in separate documents.

This is a living document and will be updated as necessary.

D.1.3 Reference Documents

[*Identify the related project documents.*]

Document	*Document Name*
High-Level Test Strategy	
Primary System Test Execution Plan	
Interfacing System Master Test Plan	
Interfacing System Test Execution Plan	
Defect and Issue Management	

© 2010 by Taylor and Francis Group, LLC

D.2 Testing Scope

[*If necessary, give an introduction on the high-level approach and guiding factors used when the testing scope was determined.*]

D.2.1 Scope Inclusions

[*Identify from a high-level point of view the features/items/requirements that will be within the scope of testing (e.g., the critical business functionalities of the core application, the application's interfaces with other applications, the performance of the application, etc.). If necessary, include a high-level diagram to provide an overview of the application. Also mention any interfaces the application has with other external systems.*]

D.2.2 Scope Exclusions

[*Identify what is out of scope for testing; for example, the full functional testing of upstream or downstream applications, failover testing, new functionality that will not be implemented, etc.*]

D.3 Roles and Responsibilities

[*Identify the project-related testing roles and responsibilities. List all responsible team and individual roles (e.g., management team, testing team, test coordinator, testing support team, etc.). This includes external/vendor teams and technical support. Each activity associated with the testing (e.g., test planning, test script creation, test script review, test data preparation, test execution, defect resolution, coordination between groups, etc.) should be listed here with the responsible group's name. If known, assigned resources with roles and responsibilities can be included. The RACI table below may be used.*]

	Project Team	*QA*	*Security*	*Business*	*<Addition>*	*<Addition>*
Documents Business and Functional requirements	A/R	C		C		
Documents Security requirements	A/R	C	R			
Create Master Test Plan	A/R	R		C		
Create Test Execution Plan	A/R	R		I		

© 2010 by Taylor and Francis Group, LLC

	Project Team	*QA*	*Security*	*Business*	*<Addition>*	*<Addition>*
Provide QA Progress Report	R	A				
Provide QA Acceptance Report	R	A				

Note: R – Responsible (performs action, assignment, task), A – Accountable (oversees task to ensure it is completed according to expected standard), C – Consulted (provides additional information and helps determine, presence required in meetings), I – Informed (provide with updates, no action required)

D.4 High-Level Test Plan

[*Identify the applicable tests that will be executed for each of the test levels (unit, integration, system, and acceptance) and provide the approach and criteria for each test. Detailed test plans for each test level or test may be documented separately. Add the document name to the Reference document list in Section 1.3.*]

D.4.1 Unit Level

D.4.1.1 Unit Testing

D.4.1.1.1 Approach

[*Describe the approach that will be taken to execute the test. Include who (or the role) will be involved in the test.*

Example: Unit testing will take place within the development phase of the project. After each application component (screen, view, interface, program, etc.) has been built to meet design specifications, it will be tested individually to help confirm that it functions properly as an individual unit. The developer will define the unit test conditions

© 2010 by Taylor and Francis Group, LLC

based on detailed design documents, and will do this testing as per these unit test conditions. The final step of the unit test will be a review by the team lead to obtain sign-off on the component test checklist.]

D.4.1.1.2 Test Environment(s) and Test Data Setup

[*Mention the test environment that will be used for testing and specific strategies, if any, for creating test user IDs, test data, and a list of the test files that will be needed, etc.*]

D.4.1.1.3 Entry/Exit Criteria

[This is a checklist. All the entry criteria listed will need to be verified for completion before beginning the testing. All the exit criteria listed need to be verified for sign-off of the unit testing. Example:

Entry Criteria	*Exit Criteria*
The module being unit tested must be completely built, configured, or set up.	Each and every unit test condition has been executed and the execution results are documented.
Unit test conditions are created and reviewed within the development team, and signed off by the developer lead.	All issues/bugs detected in testing have been documented in the defect database.
	All severity 1 issues detected have been fixed and retested successfully. The execution results of the retesting are documented.
	The execution results are signed-off by the developer lead.

D.4.1.1.4 Test Items

[*This is the list of components to be tested (e.g., screens, programs, etc.).*]

D.4.1.1.5 Test Suspension/Resumption Criteria

[*This section should answer the following questions:*
Must the entire test suite run from start to completion?
Under what circumstances may the execution of the test suite be suspended and resumed in the middle?]

© 2010 by Taylor and Francis Group, LLC

D.4.1.1.6 Risks

[*Identify the risks (if any) that are associated with this testing phase (e.g., resource constraint, availability of test environment, etc.,)*]

Risk ID	*Risk Description*	*Mitigation*
1		
2		

D.4.1.1.7 Assumptions and Dependencies

[*List the assumptions and dependencies (if any) for testing. Example: Support from Operations will be available, as and when needed, to resolve any infrastructure-related issues.*]

D.4.1.1.8 Deliverables

[*Identify the key deliverables for each test. Deliverables should be described and the source of the deliverable should be documented. Examples would be Detailed Test Schedule, Test Results, Defects Report, etc.*]

D.4.2 Integration Level Testing

D.4.2.1 Integration Testing

D.4.2.1.1 Approach

D.4.2.1.2 Test Environment(s) and Test Data Setup

D.4.2.1.3 Entry/Exit Criteria

D.4.2.1.4 Test Items

D.4.2.1.5 Test Suspension/Resumption Criteria

D.4.2.1.6 Risks

Risk ID	*Risk Description*	*Mitigation*
1		
2		

© 2010 by Taylor and Francis Group, LLC

D.4.2.1.7 Assumptions and Dependencies

D.4.2.1.8 Deliverables

D.4.3 System Level Testing

[*Identify the applicable tests that will be executed for system level testing. Possible tests are smoke, system, regression, security, systems interface, conversion, end-to-end, failover, and performance. Fill out the approach, test environment and test data setup, entry/exit criteria, etc., sections for each test that will be conducted. The security test may only be excluded if Security has signed off as "no involvement."*]

D.4.3.1 Smoke Testing

D.4.3.1.1 Approach

D.4.3.1.2 Test Environment(s) and Test Data Setup

D.4.3.1.3 Entry/Exit Criteria

D.4.3.1.4 Test Items

D.4.3.1.5 Test Suspension/Resumption Criteria

D.4.3.1.6 Risks

Risk ID	*Risk Description*	*Mitigation*
1		
2		

D.4.3.1.7 Assumptions and Dependencies

D.4.3.1.8 Deliverables

D.4.3.2 System Testing

D.4.3.2.1 Approach

D.4.3.2.2 Test Environment(s) and Test Data Setup

D.4.3.2.3 Entry/Exit Criteria

D.4.3.2.4 Test Items

© 2010 by Taylor and Francis Group, LLC

D.4.3.2.5 Test Suspension/Resumption Criteria

D.4.3.2.6 Risks

Risk ID	*Risk Description*	*Mitigation*
1		
2		

D.4.3.2.7 Assumptions and Dependencies

D.4.3.2.8 Deliverables

D.4.3.3 Regression Testing

D.4.3.3.1 Approach

D.4.3.3.2 Test Environment(s) and Test Data Setup

D.4.3.3.3 Entry/Exit Criteria

D.4.3.3.4 Test Items

D.4.3.3.5 Test Suspension/Resumption Criteria

Risk ID	*Risk Description*	*Mitigation*
1		
2		

D.4.3.3.6 Risks

D.4.3.3.7 Assumptions and Dependencies

D.4.3.3.8 Deliverables

D.4.3.4 Security Testing

[The security test may only be excluded if Security has signed off as "no involvement."]

D.4.3.4.1 Approach

D.4.3.4.2 Test Environment(s) and Test Data Setup

D.4.3.4.3 Entry/Exit Criteria

© 2010 by Taylor and Francis Group, LLC

D.4.3.4.4 Test Items

D.4.3.4.5 Test Suspension/Resumption Criteria

D.4.3.4.6 Risks

Risk ID	*Risk Description*	*Mitigation*
1		
2		

D.4.3.4.7 Assumptions and Dependencies

D.4.3.4.8 Deliverables

D.4.3.5 Systems Interface Testing

D.4.3.5.1 Approach

D.4.3.5.2 Test Environment(s) and Test Data Setup

D.4.3.5.3 Entry/Exit Criteria

D.4.3.5.4 Test Items

D.4.3.5.5 Test Suspension/Resumption Criteria

Risk ID	*Risk Description*	*Mitigation*
1		
2		

D.4.3.5.6 Risks

D.4.3.5.7 Assumptions and Dependencies

D.4.3.5.8 Deliverables

D.4.3.6 Conversion Testing

D.4.3.6.1 Approach

D.4.3.6.2 Test Environment(s) and Test Data Setup

D.4.3.6.3 Entry/Exit Criteria

D.4.3.6.4 Test Items

© 2010 by Taylor and Francis Group, LLC

D.4.3.6.5 Test Suspension/Resumption Criteria

D.4.3.6.6 Risks

Risk ID	*Risk Description*	*Mitigation*
1		
2		

D.4.3.6.7 Assumptions and Dependencies

D.4.3.6.8 Deliverables

D.4.3.7 End-To-End Testing

D.4.3.7.1 Approach

D.4.3.7.2 Test Environment(s) and Test Data Setup

D.4.3.7.3 Entry/Exit Criteria

D.4.3.7.4 Test Items

D.4.3.7.5 Test Suspension/Resumption Criteria

D.4.3.7.6 Risks

Risk ID	*Risk Description*	*Mitigation*
1		
2		

D.4.3.7.7 Assumptions and Dependencies

D.4.3.7.8 Deliverables

D.4.3.8 Failover Testing

D.4.3.8.1 Approach

D.4.3.8.2 Test Environment(s) and Test Data Setup

D.4.3.8.3 Entry/Exit Criteria

D.4.3.8.4 Test Items

© 2010 by Taylor and Francis Group, LLC

D.4.3.8.5 Test Suspension/Resumption Criteria

D.4.3.8.6 Risks

Risk ID	*Risk Description*	*Mitigation*
1		
2		

D.4.3.8.7 Assumptions and Dependencies

D.4.3.8.8 Deliverables

D.4.3.9 Performance Testing

D.4.3.9.1 Approach

D.4.3.9.2 Test Environment(s) and Test Data Setup

D.4.3.9.3 Entry/Exit Criteria

D.4.3.9.4 Test Items

D.4.3.9.5 Test Suspension/Resumption Criteria

D.4.3.9.6 Risks

Risk ID	*Risk Description*	*Mitigation*
1		
2		

D.4.3.9.7 Assumptions and Dependencies

D.4.3.9.8 Deliverables

D.4.4 Acceptance Level Testing

[*Identify the applicable tests that will be executed for acceptance level testing. Possible tests are user acceptance and operation readiness testing.*]

© 2010 by Taylor and Francis Group, LLC

D.4.4.1 User Acceptance Testing

D.4.4.1.1 Approach

D.4.4.1.2 Test Environment(s) and Test Data Setup

D.4.4.1.3 Entry/Exit Criteria

D.4.4.1.4 Test Items

D.4.4.1.5 Test Suspension/Resumption Criteria

D.4.4.1.6 Risks

Risk ID	*Risk Description*	*Mitigation*
1		
2		

D.4.4.1.7 Assumptions and Dependencies

D.4.4.1.8 Deliverables

D.4.4.2 Operational Readiness Testing

[*Failover and recovery testing should be included, if applicable.*]

D.4.4.2.1 Approach

D.4.4.2.2 Test Environment(s) and Test Data Setup

D.4.4.2.3 Entry/Exit Criteria

D.4.4.2.4 Test Items

D.4.4.2.5 Test Suspension/Resumption Criteria

D.4.4.2.6 Risks

Risk ID	*Risk Description*	*Mitigation*
1		
2		

D.4.4.2.7 Assumptions and Dependencies

D.4.4.2.8 Deliverables

© 2010 by Taylor and Francis Group, LLC

D.5 Schedule

[Include the testing schedule. The testing schedule should be as detailed as possible. It should include the start and end dates for each testing activity. If the testing schedule is updated in another file, provide the original schedule and refer to the file of the updated schedule.]

D.6 Defect Management and Resolution

[*It is recommended that statements be included that set a reasonable defect resolution time to set the expectation. For example, "The developer is required to resolve the defects within XX days."*]

The following are the severity categories being utilized by the project.

Critical	*Serious*	*Moderate*	*Minor*
Causes system to crash, abort, or freeze. Primary or major functionality is broken.	Prevents functionality of system from working and testing cannot continue (no workaround).	Results in user inconvenience and could result in some loss of functionality, but testing can continue. An acceptable workaround exists.	Results in little or no user inconvenience and no loss of functionality. Testing can proceed without interruption. An acceptable workaround exists.

[*This section should include any project-specific processes to be followed for defect management and resolution. Project-specific processes might include communicating defect lists to the vendor, escalation processes, etc.*]

D.7 Communication

[*Identify the plan for testing meetings. The QA Progress Report will be utilized for test status reporting. The frequency of the status reporting (weekly, daily) should also be noted. The QA Acceptance Report will be utilized for QA sign-off.*]

© 2010 by Taylor and Francis Group, LLC

Appendix E: New Environment Request Form (NERF)*

Contents

Revisions 184
Record of Past Modifications 184
Document Owner 184
Acronyms 184
E.1 General Project Information 185
E.1.1 Project Summary 186
E.2 Application Environment Tiers 187
E.3 Preliminary Infrastructure Impact Assessment 188
E.4 Architecture and Security 189
E.5 Implementation Details 189

* [Release Name]
New Environment Request
Version XX

© 2010 by Taylor and Francis Group, LLC

Revisions

Record of Past Modifications

Below is a record of all modifications previously made to this document.

Version	*Date*	*Name*	*Details*

Document Owner

The individual named below is the owner of this document. No modifications can be made to the document without going through this individual. All modifications made to the document will be recorded by the document owner in the "Record of Past Modifications" section above.

[Insert name of document owner]	
Job Title:	
Company:	
Dept:	
Phone:	
Mail Drop:	
Location:	
E-mail:	

Acronyms

[Add any additional acronyms as necessary.]

Acronym	*Description*
BTS	business technology solutions
CI	configuration item
CPU	central procssing unit
DFD	data flow diagram

© 2010 by Taylor and Francis Group, LLC

Acronym	*Description*
ERD	entity relationship diagram
GB	gigabyte
LAN	local area network
MTP	master training plan
OR	Operational Readiness
SIP	standard investigative procedures
SLA	service level agreement
SLM	service-level management
SLO	service level objective
WAN	wide area network
	<insert any additional acronyms as needed>

E.1 General Project Information

Name of project	[Enter text here.]	
Approved: Yes/No	***Approval Score:***	[Enter score]
Current project	Planning, Design, Build or Test	
Business unit	Business Unit	
Project manager	PM If Assigned	EXT.:
Project sponsor	[Enter name here]	EXT.:
Project tech lead	[Enter name here]	EXT.:
Assigned architect	[Enter name here]	EXT.:
Date submitted to design and engineering	mm-dd-yy	
Target completion date	Please list target completion dates BY ENVIRONMENT (See Section 2.0 to describe environments)	
Project cost center	Unless specified, no hardware or software will be procured	
Project ID	Unless specified, no hardware or software will be procured	
Department ID	Unless specified, no hardware or software will be procured	

© 2010 by Taylor and Francis Group, LLC

E.1.1 Project Summary

Please provide a summary of the project in the space below. Include a brief explanation of the current situation, the planned project, the new process, and the expected business benefit.

Current situation:
[Enter text here]
Planned project:
[Enter text here]
New process:
[Enter text here]
Expected business benefit:
[Enter text here]

Please fill in the information or check the appropriate box.

Project Type		
() Budgeted	Capital Hardware $	
() Nonbudgeted	Capital Software $:	

High-Level Scope *(Please check known technology areas that this project will affect.)*	
() Enterprise Support Services (Help Desk)	() Mainframe
() Enterprise Support Services (Deskside)	() Field Location
() Telecom/Voice	() Other (Please List)
() Customer Service Center	
() LAN Services	
() WAN Services	
() UNIX® Services	
() Windows® Services	

© 2010 by Taylor and Francis Group, LLC

E.2 Application Environment Tiers

Tier	*Description*	*Recovery Point Objective (RPO)*	*Return to Operations (RTO)*
1	Continuous operation	<insert specification>	<insert specification>
2	Near continuous	<insert specification>	<insert specification>
3	Three-day recovery	<insert specification>	<insert specification>
4	Noncritical	<insert specification>	<insert specification>

Note: These tiers only apply to production environments.

Based on the information located above, please answer the following questions.

1	What is the expected tier of the production application environment?	*Note*: Tier 1 applications will have to undergo initial and regular certification tests in order to maintain that status.
2	Which environments are needed to support the project?	() Development () Test () Stage () Training () Production
3	In the event of a disaster, how long can business sustain operations without this application?	Please be specific; for example, for each hour the application is unavailable, revenue is affected by $ xx,000.
4	Please outline the business continuity plan for the functionality.	In what location will the business need to operate the application in the event that primary sites are unavailable?

© 2010 by Taylor and Francis Group, LLC

E.3 Preliminary Infrastructure Impact Assessment

Please provide any known information about the technologies that will be deployed. Note: It is understood that this information may change as the project progresses.

1	Specify the targeted operating platform(s) for the project's technology. Check all that apply.	() UNIX (Solaris) () Windows () Desktop (Proprietary software client) () Desktop (Web browser access only)
2	Is this a software package implementation or a software development project?	For a package implementation, specify the package and vendor. If the selection process is not complete, specify the candidate vendors. For a software development project, specify the development environment and any development partners.
3	Is any technology or system being *completely* replaced or phased out?	() Yes () No If so, identify the system(s):
4	Identify all known systems with which the new technology must interface.	
5	Estimate the number of users affected by the implementation.	If known, differentiate among specific locations.
6	Are any hardware requirements or architectural recommendations already known? Identify environment and attach details.	If known, specify details (e.g., environments, CPU requirements, memory requirements, disk requirements) for each: CPU Speed (MHz): Disk Space (local, in GB):
7	Does this project have offsite build, operation, or maintenance requirements?	
8	Does this technology require support outside of the standard operating hours? If so, please explain.	

© 2010 by Taylor and Francis Group, LLC

E.4 Architecture and Security

Please provide knowledge of how this project will map into the existing organizational architecture and security.

Note: Information technology is responsible for maintaining consistency in security and architecture of technology projects.

1	Does this project's architecture match one of the published standard architectures?	*Projects that do not will be reviewed by the Architectural Review Committee.*
2	Does this project require the acquisition of software?	*Software acquisitions and external providers must comply with architectural standards and contract requirements.*
3	Has this project been brought to the attention of the IT Security Team?	

E.5 Implementation Details

Please provide knowledge of how this project will be implemented.

Note: Information technology is responsible for maintaining consistency in security and architecture of technology projects.

[Enter text here]

© 2010 by Taylor and Francis Group, LLC

Appendix F: Infrastructure Design Document (IDD)*

Contents

Revisions 192
Record of Past Modifications 192
Document Owner 192
Design Approval 193
Approval to Procure 193
Acronyms 194
F.1 General System Design 194
F.2 Configuration Items (CI) 194
F.2.1 [Configuration Item Name] 195
F.2.2 [Configuration Item Name] 195
F.2.3 [Configuration Item Name] 195
F.2.4 [Configuration Item Name] 196
F.2.5 [Configuration Item Name] 196
F.2.6 [Configuration Item Name] 196
F.2.7 [Configuration Item Name] 197
F.2.8 [Configuration Item Name] 197

* [Project Name]
Information Technology Services Projects
Infrastructure Design Document
Version XX

© 2010 by Taylor and Francis Group, LLC

F.2.9 [Configuration Item Name] 197
F.2.10 [Configuration Item Name] 198
F.3 Risk and Assumptions 198
F.4 Cost Estimation 198
F.5 Resource Estimation 199

Revisions

Record of Past Modifications

Below is a record of all modifications previously made to this document.

Version	*Date*	*Name*	*Details*

Document Owner

The individual named below is the owner of this document. No modifications can be made to the document without going through this individual. All modifications made to the document will be recorded by the document owner in the "Record of Past Modifications" section above.

[Insert document owner]	
Job Title:	
Company:	
Dept:	
Phone:	
Mail Drop:	
Location:	
E-mail:	

© 2010 by Taylor and Francis Group, LLC

Design Approval

Your approval indicates that you have reviewed the initial infrastructure design for the subject environment and agree with the proposed design.

Project Owner

Project Manager

Architecture

[Involved Infrastructure team, e.g., Database] Infrastructure Team

[Involved Infrastructure team] Infrastructure Team

[Involved Infrastructure team] Infrastructure Team

[Involved Infrastructure team] Infrastructure Team

[Involved Infrastructure team] Infrastructure Team

Information Technology Security Group

Design & Engineering

Release Management

Approval to Procure

Your approval indicates that you have reviewed the initial infrastructure design for the subject environment and agree with the proposed design. Your approval also gives your approval to procure the hardware and software.

Chief Technology Officer

© 2010 by Taylor and Francis Group, LLC

Acronyms

Please add any additional acronyms as necessary.

Acronyms	*Description*
CI	configuration item
DBA	database administrator
ORT	operational readiness testing
SLA	service level agreement
[additional]	
[additional]	
[additional]	

F.1 General System Design

Describe the system requirements, operating environment, system and subsystem architecture, files and database design, input formats, output layouts, human–machine interface, detailed design, processing logic, and external interfaces. Also include context diagrams, dataflow diagrams, and physical infrastructure diagrams.

[Enter text and/or diagrams here.]

F.2 Configuration Items (CI)

Expand on the general design by providing all the specific technical details of each configuration item. Configuration items include information such as:

- **Purpose** (scope or targeted goal)
- **Functional Design** (business rules, application logic, or security roles)
- **Technical Design** (hardware, database, network, firewalls)
- **Input** (user interface, app-to-app interface, app-to-database interface)
- **Processing** (accessing and processing data)
- **Output** (user interface, database-to-app interface, utilization of data)

© 2010 by Taylor and Francis Group, LLC

F.2.1 [Configuration Item Name]

Purpose
[Enter text and/or diagrams here.]
Functional Design
[Enter text and/or diagrams here.]
Technical Design
[Enter text and/or diagrams here.]
Inputs
[Enter text and/or diagrams here.]
Processing
[Enter text and/or diagrams here.]
Outputs
[Enter text and/or diagrams here.]

F.2.2 [Configuration Item Name]

Purpose
[Enter text and/or diagrams here.]
Functional Design
[Enter text and/or diagrams here.]
Technical Design
[Enter text and/or diagrams here.]
Inputs
[Enter text and/or diagrams here.]
Processing
[Enter text and/or diagrams here.]
Outputs
[Enter text and/or diagrams here.]

F.2.3 [Configuration Item Name]

Purpose
[Enter text and/or diagrams here.]
Functional Design
[Enter text and/or diagrams here.]
Technical Design
[Enter text and/or diagrams here.]
Inputs
[Enter text and/or diagrams here.]
Processing
[Enter text and/or diagrams here.]
Outputs
[Enter text and/or diagrams here.]

© 2010 by Taylor and Francis Group, LLC

F.2.4 [Configuration Item Name]

Purpose
[Enter text and/or diagrams here.]
Functional Design
[Enter text and/or diagrams here.]
Technical Design
[Enter text and/or diagrams here.]
Inputs
[Enter text and/or diagrams here.]
Processing
[Enter text and/or diagrams here.]
Outputs
[Enter text and/or diagrams here.]

F.2.5 [Configuration Item Name]

Purpose
[Enter text and/or diagrams here.]
Functional Design
[Enter text and/or diagrams here.]
Technical Design
[Enter text and/or diagrams here.]
Inputs
[Enter text and/or diagrams here.]
Processing
[Enter text and/or diagrams here.]
Outputs
[Enter text and/or diagrams here.]

F.2.6 [Configuration Item Name]

Purpose
[Enter text and/or diagrams here.]
Functional Design
[Enter text and/or diagrams here.]
Technical Design
[Enter text and/or diagrams here.]
Inputs
[Enter text and/or diagrams here.]
Processing
[Enter text and/or diagrams here.]
Outputs
[Enter text and/or diagrams here.]

© 2010 by Taylor and Francis Group, LLC

F.2.7 [Configuration Item Name]

Purpose
[Enter text and/or diagrams here.]
Functional Design
[Enter text and/or diagrams here.]
Technical Design
[Enter text and/or diagrams here.]
Inputs
[Enter text and/or diagrams here.]
Processing
[Enter text and/or diagrams here.]
Outputs
[Enter text and/or diagrams here.]

F.2.8 [Configuration Item Name]

Purpose
[Enter text and/or diagrams here.]
Functional Design
[Enter text and/or diagrams here.]
Technical Design
[Enter text and/or diagrams here.]
Inputs
[Enter text and/or diagrams here.]
Processing
[Enter text and/or diagrams here.]
Outputs
[Enter text and/or diagrams here.]

F.2.9 [Configuration Item Name]

Purpose
[Enter text and/or diagrams here.]
Functional Design
[Enter text and/or diagrams here.]
Technical Design
[Enter text and/or diagrams here.]
Inputs
[Enter text and/or diagrams here.]
Processing
[Enter text and/or diagrams here.]
Outputs
[Enter text and/or diagrams here.]

© 2010 by Taylor and Francis Group, LLC

F.2.10 [Configuration Item Name]

Purpose
[Enter text and/or diagrams here.]
Functional Design
[Enter text and/or diagrams here.]
Technical Design
[Enter text and/or diagrams here.]
Inputs
[Enter text and/or diagrams here.]
Processing
[Enter text and/or diagrams here.]
Outputs
[Enter text and/or diagrams here.]

F.3 Risk and Assumptions

Risk Number	*Risk/Assumption*	*Mitigation/Risk*

F.4 Cost Estimation

The costs of the required hardware and software are estimated to be as follows:

CI	*Description*	*Amount*

© 2010 by Taylor and Francis Group, LLC

CI	*Description*	*Amount*
	Total	

F.5 Resource Estimation

Description	*Resource*	*Timeframe (hours)*	*Rate*	*Cost*
Total				

© 2010 by Taylor and Francis Group, LLC

Appendix G: Security Risk Assessment*

Contents

Revisions 201
Record of Past Modifications 201
Document Owner 202
Acronyms 202
G.1 Application or Infrastructure Overview 203
G.1.1 Overview of Application or System 203
G.1.2 Business Process Diagram 203
G.1.3 Network Diagram 203
G.2 Project Detail Questions 203

Revisions

Record of Past Modifications

Below is a record of all modifications previously made to this document.

Version	*Date*	*Name*	*Details*

* [Release Name]
Information Technology Services Projects
Security Risk Assessment Questionnaire
Version X.X

© 2010 by Taylor and Francis Group, LLC

Document Owner

The individual named below is the owner of this document. No modifications can be made to the document without going through this individual. All modifications made to the document will be recorded by the document owner in the "Record of Past Modifications" section above.

[insert document owner name]	
Job Title:	
Company:	
Dept:	
Phone:	
Mail Drop:	
Location:	
E-mail:	

Acronyms

Please add any additional acronyms as necessary.

Acronyms	*Description*
ASP	application service provider
B2B	business-to-business
B2C	business-to-customer
B2E	business-to-enterprise
CI	configuration item
ESS	enterprise support services
MTP	master training plan
OR	operational readiness
ORT	operational readiness testing
SLA	service level agreement
SLM	service-level management
SLO	service level objective
SIP	standard investigative procedures
	<insert any additional>

© 2010 by Taylor and Francis Group, LLC

G.1 Application or Infrastructure Overview

G.1.1 Overview of Application or System

Please describe at a high level the purpose of the application or system to be used within this project.

[Enter overview here.]

G.1.2 Business Process Diagram

A business process diagram is a high-level description showing how the information will flow. The diagram must identify the major components of the business process and how information is collected, how it circulates within the organization or any third-party organization, and how it will be used and where it will be retained.

[Enter business process diagram here.]

G.1.3 Network Diagram

A network diagram shows how network equipment is logically connected and where within the network everything resides. It also shows how the network equipment is physically connected or arranged at a specific site, such as a server room or data center. Please indicate in the diagram any ports, protocols, and services used within the applications or infrastructure. Please be sure to include host names and platforms, including hardware configurations, operating system, and patch and service pack levels associated with each host and application.

[Enter network diagram here.]

G.2 Project Detail Questions

Below are general questions that will be used by the Infrastructure Technology Security Office to get a better understanding of the overall project and to drive the formal security requirements for the project.

1. The information owner is the individual designated by management of the applicable business unit to be the person ultimately responsible for the security and privacy of particular information or information assets. Please indicate who the designated information owner will be.

Information Owner:	

2. Will the system(s) in any way store or process information classified as confidential, protected, or public? Please explain what information is being stored or processed. *Note*: All customer or consumer information is classified as

© 2010 by Taylor and Francis Group, LLC

confidential information. If nonpublic information has been identified, please include your reasoning as to why you feel it should be classified as such.

Note: To be completed by information owner.

() Confidential () Protected () Public () Not sure

[Enter rationale for classification here.]

3. Will the system(s) in any way store or process customer or consumer information?

 () Yes () No () Not sure

4. Will the system(s) contribute materially to financial reporting under the Sarbanes–Oxley Act?

 () Yes () No () Not sure

5. Will the system(s) reside in the business-to-business (B2B) space (e.g., dealers), business-to-consumer (B2C) space (e.g., customers or consumers), or business-to-enterprise (B2E) space (e.g., internal workers only)?

 () B2B () B2C () B2E (Internal Only) () Not sure

6. Will the system(s) be accessed by any other affiliates?

 () Yes () No () Not sure

7. Will the development of the system(s) be done internally or is the system(s) a commercial, off-the-shelf product developed by a third-party vendor?

 () Internally developed () Purchased product () Not sure

8. Will the system(s) reside internally or be externally hosted and managed by an outside party (e.g., ASP model)?

 () Internal () External () Combination of both () Not sure

9. Please list any physical, technical, or administrative control measures that are already in place or that will be implemented to protect the confidentiality, integrity, and availability of the information that will be stored or processed within the system(s). Will these controls be monitored and regularly tested? If so, how often are the controls monitored and tested?

 [Enter description here.]

© 2010 by Taylor and Francis Group, LLC

10. Is there a Web-based application that is in the scope of this project? If so, please describe the architecture and platform that will be used to host the Web application. Please also describe the functions that it will provide.

 () Yes () No () Not sure

 [Enter description here.]

11. Will the system(s) have its own built-in authentication and authorization process? Will the system(s) manage its own user IDs internally? If so, please indicate how the user IDs and any associated roles will be managed.

 () Yes () No () Not sure

 [Enter description here.]

12. Will there be a need to have different levels of access within the system(s)? Please describe. Please also describe any superuser, administrative-type user, or role and who will have access to that role.

 () Yes () No () Not sure

 [Enter description here.]

13. Please describe the overall envisioned access approval workflow process from initial request to final approval that will be in place to grant individuals access to the system(s). Please be sure to include who will be responsible for reviewing and approving access.

 [Enter description here.]

14. Will there be any operational batch or online processes used to support the system(s)? If yes, please describe these processes including the type of information and high-level data flow.

 () Yes () No () Not sure

 [Enter description here.]

15. If a third party is to be used for this project, please describe if any data is expected to be transferred. Please also indicate what exactly will be included in the transfer and how often this data will be transferred.

 [Enter description here.]

16. Please list any third parties that will be used during this project.

 [Enter description here.]

© 2010 by Taylor and Francis Group, LLC

17. If a third party is anticipated to be used, has a contract been signed already with the third party?

 () Yes () No () Not sure

 [Enter description here.]

18. Will the third party require inbound access to the internal network? If so, please indicate what will need to be accessible and the business justification. Has a Third Party Access Agreement been established, reviewed, and approved by management?

 () Yes () No () Not sure

 [Enter description here.]

19. Will there be any anticipated reporting (transactional, management, audit, regulatory) available within the system(s)? Please describe the overall review processes. How often would this be performed and who would have the responsibility for this task?

 [Enter description here.]

20. If external hosting is involved for the system(s), is there an incident investigation and notification process in place and formally documented?

 () Yes () No () Not sure

 [Enter description here.]

21. Must any component of the system(s) install or run under any privileged accounts? This would include administrative accounts such as "root" or "administrator."

 () Yes () No () Not sure

22. Does documentation exist that describes all default IDs in the system(s)?

 () Yes () No () Not sure

23. Can all default IDs in the system(s) be either disabled, removed, or have their passwords changed?

 () Yes () No () Not sure

24. If the system(s) is hosted, list all ports and protocols that are used.

 [Enter description here.]

25. If the system(s) requires a connection to a third party, describe how the third party manages vulnerabilities.

 [Enter description here.]

© 2010 by Taylor and Francis Group, LLC

26. Will the system(s) be hosted internally and require any third-party remote access or monitoring for its operation?

 () Yes () No () Not sure

27. If the system(s) is hosted externally, is access to the production environment limited to administrative personnel only?

 () Yes () No () Not sure

© 2010 by Taylor and Francis Group, LLC

Appendix H: Functional Specifications*

Contents

Revisions 210
Record of Past Modifications 210
Document Owner 210
Abbreviations 211
Involved Infrastructure and Network Teams 212
H.1 Overview 212
H.2 Contents 212
H.3 Roles and Responsibilities 213
H.4 General Requirements 213
H.4.1 Supporting Documents 213
H.5 Database Request 213
H.5.1 Database Description 213
H.5.2 Location of Database (Server) 214
H.5.3 Type and Version of Database 214
H.5.4 Date Required 214
H.6 Integration Request 214
H.6.1 Interface Technology 214
H.6.2 Transport 214
H.6.3 Data Format 214
H.6.4 Communication Mode 214
H.6.5 Error Correction and Recovery 215
H.6.6 Interface Description 215

* [Release Name]
Information Technology Services Projects
Functional Specifications
Version x.x

© 2010 by Taylor and Francis Group, LLC

H.7 Network Request 215
H.7.1 User Information 215
H.7.2 Network Environment 215
H.7.3 Load Balancing 215
H.7.4 Network Security 215
H.8 Security Request 216
H.8.1 IT Security 216
H.8.2 Security Access Integration 217
H.9 Storage Request 219
H.9.1 Storage Area Network (SAN) Policy 219
H.9.2 Type of Information to be Stored 219
H.9.3 Primary Storage 219
H.9.4 Storage Connections 220
H.9.5 Disk Space 220
H.9.6 Application Details 220
H.9.7 Storage Retirement 220
H.10 Web Services Request 221
H.10.1 Web Server 221
H.10.2 Application Server 221
H.10.3 Content Management 223
H.10.4 Security/SSO 223
H.10.5 Performance 223
H.10.6 Other 224
Attachment A: Roles and Responsibilities 224

Revisions

Record of Past Modifications

Below is a record of all modifications previously made to this document.

Version	*Date*	*Name*	*Details*

Document Owner

The individual named below is the owner of this document. No modifications can be made to the document without going through this individual. All modifications made to the document will be recorded by the document owner in the "Record of Past Modifications" section above.

© 2010 by Taylor and Francis Group, LLC

[Insert document owner's name and info]	
Job Title:	
Company:	
Dept:	
Phone:	
Mail Drop:	
Location:	
E-mail:	

Abbreviations

Acronyms	*Description*
API	application programming interface
CPU	central processing unit
DTD	data transfer device
GUI	graphical user interface
IDD	infrastructure design document
JDBC	Java database connectivity
JMS	Java machine service
JVM	Java virtual machine
NER	New Environment Request (phase of the RLC)
NSL	neural simulation language
OBDC	open database connectivity
RFC	request for change
RLC	release lifecycle
RACI	responsible, accountable, consulted, and informed
RM	release management
SAN	storage area network
SQL	structured query language
SSL	secured socket layer
SSO	single sign-on
UAT	user acceptance testing

© 2010 by Taylor and Francis Group, LLC

Involved Infrastructure and Network Teams

Please indicate the teams that are involved with this request: (These are the sections of the document that need to be completed.)

- ☐ Database
- ☐ Integration
- ☐ Network
- ☐ Security—**Required**
- ☐ Storage
- ☐ Web Services

H.1 Overview

The purpose of this document is to supplement the New Environment Request (NER) process. The Functional Specification Document provides basic requirements that are used by the respective infrastructure teams to design and provide environment requirements to the architect. The assigned architect will take the requirements provided and incorporate them into the infrastructure design document (IDD). It is important that the requirements provided are accurate so the infrastructure teams and architect can provide an environment that provides serviceability, maintainability, reliability, and scalability to meet the requirements and needs of the project objectives.

H.2 Contents

The Functional Specifications form is a combined form that is used by database, integration, security, storage, Web services, and network groups to understand the requirements of the environment requested. The document combines these forms to help streamline the process and allows the requester to provide basic project information once instead of several times for individual documents.

There are two types of information needed within the document: required and "as needed" information. The required fields are basic project information that will be used by all of the infrastructure teams and are needed to understand the project. The "as needed" fields only need to be completed if the respective team is needed to complete the design.

© 2010 by Taylor and Francis Group, LLC

H.3 Roles and Responsibilities

It is the responsibility of the project manager to coordinate with the project team, the architect, and the infrastructure team to complete the document and provide it to release management. Release management will then forward it to the respective teams for review and design of the environment. A RACI (responsible, accountable, consulted, and informed) matrix is attached to help understand these roles and responsibilities (see Attachment A).

H.4 General Requirements

The general description and purpose of the application or system being developed can be found in the New Environment Request Form. This document should be referenced to gain an understanding of the goals and objectives of the project.

H.4.1 Supporting Documents

In order to assist participants in gaining a full understanding of the project, specific documents, if available, should be attached to this Functional Specification document. These documents will enable the teams to provide more meaningful design recommendations. The types of documents that should be attached include, but are not limited to, business process diagrams, system flow documents, and network diagrams.

H.5 Database Request

H.5.1 Database Description

Describe the database you wish to have created, including the server and version information, and when it needs to be created. Enter as much detailed information as possible. Information should include the database schema, number of tables required, specific field requirements, application or applications that need to interface with the database workflows, workflows the database needs to accommodate, size requirements, character set, and neural simulation language (NSL) language that requires any specific new feature. If this is a vendor package, indicate the vendor's name. If a homegrown system, specify and attach a logical diagram.

© 2010 by Taylor and Francis Group, LLC

H.5.2 Location of Database (Server)

If the database needs to reside on a predetermined server, please specify. May not be applicable if this is a new requirement, we will then dictate where the DB should go.

H.5.3 Type and Version of Database

Enter the type of database you want to be created. For example, Oracle 10g database, SQL, Access, etc.

H.5.4 Date Required

Indicate the date or deadline for creation of the new database.

H.6 Integration Request

H.6.1 Interface Technology

What technologies does the application support for interfacing with external systems? (Off-the-shelf technologies as part of the base package? As a separate licensed component?) Does the application provide interface objects/configurations specifically compatible with a vendor's implementation?

H.6.2 Transport

What physical data transport mechanisms does the application support? What transport protocols does the application support? Batch transfers (if applicable)? Event-based or request/response transfers (if applicable)?

H.6.3 Data Format

What file formats are supported for file transfers (if applicable)? Is a data transfer device (DTD) or schema available for XML messages (if applicable)?

H.6.4 Communication Mode

What modes of communication does the application support? Request/response, asynchronous message, fire and forgot, publish and subscribe, broadcast, or other mechanisms of communication? How does the application enforce communication with external applications via interfaces?

© 2010 by Taylor and Francis Group, LLC

H.6.5 Error Correction and Recovery

Does the application expose interface errors to external applications? How does the application manage interface business or runtime errors?

H.6.6 Interface Description

Include a data flow diagram and interface matrix clearly denoting the following: application sources, application interface types (Java message service [JMS], Web service, database-to-database, Oracle DB, file), application interface server locations, one-to-many and many-to-one interface interactions, interface mode (sync request/response, fire and forget, batch), interface response time critical to requirements, as-is and to-be diagrams (if existing interfaces will be reworked).

H.7 Network Request

H.7.1 User Information

Who are the users, how many users are there, and where are they located?

H.7.2 Network Environment

How is the application architected (i.e., fat client, Web enabled)? Will the application be in different environments (Dev, Stage, Prod)? What other applications, business partners, connections are needed? What is the required latency in milliseconds (i.e., does it require voice quality or can it retransmit)?

H.7.3 Load Balancing

☐	Tier 1	Globally load balanced
☐	Tier 2	Local load balancing
☐	Tier 3	No load balancing
☐	Tier 4	Doesn't matter

H.7.4 Network Security

What level of security is necessary? Is data to be encrypted? What is the source destination?

© 2010 by Taylor and Francis Group, LLC

H.8 Security Request

The following are general questions that will be used by Information Technology Security to get a better understanding of the overall project and to drive formal security requirements.

H.8.1 IT Security

1. Will the development of the system(s) be done internally or is the system(s) a purchased commercial off-the-shelve product or some combination?

 ☐ Internally Developed ☐ Purchased ☐ Combination ☐ Not Sure

2. Will the system(s) reside internally or be externally hosted and managed by an outside party (i.e., ASP model)?

 ☐ Internal ☐ External ☐ Combination of Both ☐ Not Sure

3. Will the system(s) have its own built-in authentication process? Will the system(s) manage its own user IDs internally? If so, describe how the user IDs and any associated roles will be managed.

4. Describe any required service accounts including privileged "superuser" or administrative accounts.

5. Will any third parties require access to data? If so, name each third party and describe the type of data.

 ☐ During the project only
 ☐ For production operations only
 ☐ Both
 ☐ Not sure
 ☐ Inbound transfers only
 ☐ Outbound transfers only
 ☐ Both inbound and outbound transfers

6. Will the system(s) be hosted internally and require any third-party remote access or monitoring for its operations? If yes, please describe.

 ☐ Yes ☐ No

© 2010 by Taylor and Francis Group, LLC

7. Must any component of the system(s), either server-side or client-side, install or run under any privileged accounts? This would include administrative accounts such as "root" or "administrator." If so, please describe.

 ☐ Yes ☐ No ☐ Not Sure

8. Does documentation exist that describes all default IDs in the system(s)?

 ☐ Yes ☐ No ☐ Not Sure

9. Can all default IDs in the system(s) be disabled, removed, or have their passwords changed?

 ☐ Yes ☐ No ☐ Not Sure

10. If the system(s) is hosted externally, can IT freely apply software patches at its sole discretion?

 ☐ Yes ☐ No ☐ Not Sure

H.8.2 Security Access Integration

1. Please describe the ID creation and ID administration process. Please provide approval details on how to request access to this application. What is the process for adding new users to the application?

2. Describe the overall envisioned access approval workflow process for initial request to final approval that will grant individuals access to the system(s). Include the operations roles that will be responsible for reviewing and approving access.

3. Please list the different system roles (also known as groups, access level, or responsibilities).

4. What type of data exchange is available? Is there any Java API that could be used for ID creation/modification/disablement/enablement/deletion?

5. In a case where IDs are directly on the database, not through the application, could Java database connectivity (JDBC) or open database connectivity (ODBC) be used? How many tables are involved? Can the tables be accessed directly?

© 2010 by Taylor and Francis Group, LLC

6. Please describe in detail the user-specific information that is in the database (e.g., name, manager, phone, mail stop, etc.)

7. What type of operations and input parameters are involved when an ID is created, responsibility modified, or an ID is disabled, enabled, and deleted?

8. Is there any additional operation for adding IDs and/or responsibilities for complete ID setup? If so, please explain.

 ☐ Yes ☐ No

9. Please describe the process of changing or adding the responsibility for an existing user.

10. Is there any history table maintained as a part of ID updates? If yes, please describe.

 ☐ Yes ☐ No

11. Is the reconciliation with security access required? Is a modification necessary in any of the lookup tables?

 ☐ Yes ☐ No

12. After user creation (add, update, or delete) is there any workflow required (e.g., e-mail needs to be sent)? If yes, please explain.

 ☐ Yes ☐ No

13. Is there any ID name–generation rule? If yes, what is the user ID nomenclature?

 ☐ Yes ☐ No

14. Are any password policies enforced? If yes, please explain.

 ☐ Yes ☐ No

15. Is there any limitation on the number of connections? Is the number of sockets limited? If yes, please explain.

 ☐ Yes ☐ No

16. How many environments are there for this application? What are the environments?

© 2010 by Taylor and Francis Group, LLC

17. Is this is a Web application, and is SSO required? If yes, what are the Web servers' names and locations? Does the application natively support SSO with windows (Windows® authentication)? If yes, please describe.

 ☐ Yes ☐ No

18. Can the system use Active Directory as the user store?

 ☐ Yes ☐ No

19. Can the system use Active Directory groups as the source of user roles?

 ☐ Yes ☐ No

20. What is the name of each environment application URL?

21. How many users per environment?

H.9 Storage Request

H.9.1 Storage Area Network (SAN) Policy

Minimum space requirements for requesting addition to the SAN:
■ UNIX® systems: > xx GB of storage ■ NT systems: > xxx GB of storage
SAN Port Standards:
■ Development or Staging: x ■ Production: x

H.9.2 Type of Information to be Stored

What type of information will be stored? What level is the business criticality of this information?

H.9.3 Primary Storage

Is the information to be stored primarily transactional or reporting?

☐ - Transactional ☐ - Reporting

© 2010 by Taylor and Francis Group, LLC

H.9.4 Storage Connections

Which environments will the storage need to be connected to? How many connections are needed and how long?

Environment	*QTY*	*Start Date*	*Duration*
☐ Development			
☐ Staging			
☐ Production			
☐ Test			

H.9.5 Disk Space

How much disk space will be required when first migrating this data to the storage area? What are the predicted storage needs over the next 6 to 12 months and over the long term? (Work with the assigned project database administrator to determine this estimate.)

Production	Disk Space Required	
	Growth Assumption	
	Growth % (per annum)	
It will be assumed that the disk requirements for the development and staging environments are the same as the requirements for the production environment (x quantity) unless specified differently below.		
Development	Disk Space Required	
Staging	Disk Space Required	
Test	Disk Space Required	

H.9.6 Application Details

What applications require the storage of this data? What is the purpose of the applications? What are the primary functions of the applications? What other applications are affected? Are these applications dependent on other applications?

H.9.7 Storage Retirement

Is there any storage being retired as a result of this request? If yes, which ones and where do they currently reside?

© 2010 by Taylor and Francis Group, LLC

Name of Database	Location	Server Name	IP Address	Retirement Date

H.10 Web Services Request

H.10.1 Web Server

Please supply the following information as known.

- Which Web server and version is needed?
- On which port will the Web server listen?
- On which port will the admin server listen?
- Who will need access to the administration GUI? Please provide name and physical location.
- Please specify the conditions in which the Web server should proxy the request to the application server. (Example: Should all requests be proxied to Weblogic or only to specific URL patterns?)
- Will the Web server host any content?

H.10.2 Application Server

Please supply the following information as known.

- What kind of installation is desired?
- What kind of domain is required for the project? Server domain; platform domain; portal domain?
- What Java virtual machine (JVM) is the application certified on?
- Will this run on an existing instance or a new instance?
 - If a new instance, can the application use a standard install? Or, are customizations required (for example, fix-packs, classpath)?

© 2010 by Taylor and Francis Group, LLC

- Are there known/recommended sizing specifications (number of CPUs, CPU speed, minimum memory requirements on server, storage requirements [SAN required?])?

- Is logging required?
 - Does it use log4j? If so, provide the following answers:
 - What kind of log rotation is required?
 - How will the files be archived?
 - Does the log file have to be moved to a new server after it is rotated?
 - How will log files be accessed and by whom (include location)?

- What kind of availability is required? What is the failover scenario(s)? Is clustering of Web servers or app servers required?

- What database is used?
 - Will container-managed transaction be used? Does the driver need to support transaction?
 - Which database driver do you require?
 - Does the application use any specific connection pool name or data source?

- Resource configuration
 - What are the resources that need to be configured (e.g., JMS, JDBC, messaging bridge, XML registries)?

- Security
 - Will default security providers be used?

- Deployment
 - Will an external tool be used for deployment? If not, will the application deployments be scripted?

- Monitoring
 - Are there special monitoring requirements?

- Licensing (these questions are relevant for nonproduction environments)
 - Does the server need to be in production mode?
 - Will all domains on the servers be in production mode?
 - Do you require more than five different users/developers to access the application (specify per environment—dev/test/etc.)?
 - Who pays for the license? Does it come out of the project budget?

What applications, systems, or back-ends does this Web application communicate with? Per communication, please answer the following:

© 2010 by Taylor and Francis Group, LLC

- For the connection pool, how many connections are needed for the initial deployment? (Note: The final number will be decided based on the results of the load/performance tests.)

- What protocol will be used for communication (JDBC, Web services, JMS, etc.) and on what port(s)?

On which port will the server listen?

Who will need access to the administration GUI? Please provide names, LAN ID, physical location, and the roles that are required for the LAN IDs (e.g., the different roles that are available are admin, monitor, deployer, and operator).

H.10.3 Content Management

Is content management required for the application; i.e., will business users be updating content?

H.10.4 Security/SSO

Please supply the information as known.

Will the Web application be secure (https://) or nonsecure (http://)?

- Will SSL be limited to the Web server or is secure socket layer (SSL) required?
- What percentage of transactions will be SSL enabled?

Will the built-in authentication mechanism be used or a will a third-party authentication system be integrated with the portal?

Will SSO be configured?

H.10.5 Performance

Please provide the information as known.

How many concurrent users are you expecting on the system under normal circumstances?

How many concurrent users are you expecting on the system during activity spikes (for example, at the end of the month or end of the quarter, special promotions, etc.)?

What is the min, max, and average time that a transaction takes when interacting with back-end processes exposed?

© 2010 by Taylor and Francis Group, LLC

What is the min, max, and average time that a transaction takes when interacting with databases?

What is the highest tolerable response time for the portal home page?

What is the highest tolerable response time for the log-in process?

What is the highest tolerable response time for the pages that have portlets with personalized content?

H.10.6 Other

Will this be used internally, externally, or both? Does the project require any development tools? Is a licensing server needed (new vs. existing)? Is there any other relevant information to determine the application characteristics?

Attachment A: Roles and Responsibilities

	Project Manager	*Architecture*	*Infra Team*	*Project Team*	*Release Mgt*
Originate functional specification	A/R	C	C	C	
Review completed documentation	C		A/R		R
Create functional design	C	C	A/R	I	
Utilize functional specifications and complete IDD	C	A/R	C	I	I

Note: A – Accountable (oversees task to ensure it is completed according to expected standard), R – Responsible (performs action, assignment task), C – Consultative (provides additional information and helps determine, presence required in meetings), I – Informed (provide with updates, no action required)

© 2010 by Taylor and Francis Group, LLC

Appendix I: Operational Readiness Testing Manual (ORT)*

Contents

Revisions 226
Record of Past Modifications 226
Document Owner 226
Operational Readiness Testing (ORT) Completion Approval 227
Completion Certification 227
Abbreviations 228
I.1 Overview 229
I.2 ORT Process 230
I.2.1 ORT Preparation 230
I.2.1.1 Summary Artifact 230
I.2.2 Preprocess 231
I.2.3 ORT Steps 231
I.2.4 Updating the CMDB 232
I.3 ORT Checklist 233
I.3.1 Common Task List 233
I.3.2 Tools Acceptance Task List 234
I.3.3 Monitoring Tools Acceptance Task List 235
I.3.4 Server QA Checklist for Windows® Operations 235
I.3.5 Backup Acceptance Task List 237
I.3.6 Checklist for Metaframe Servers 238

* [Release Name]
Information Technology Services Project
Operations Readiness Testing Manual
Version [x.x]

© 2010 by Taylor and Francis Group, LLC

I.3.7 UNIX® Acceptance Task List 238
I.3.8 SAN Storage Checklist 240
I.3.9 Database Configuration Checklist 241
I.3.10 CMDB Configuration Checklist 249
Attachment A: Roles and Responsibilities 250
Operational Readiness Testing 250
Attachment B: Key Application Information Guide 251
Revisions 251
Record of Past Modifications 251
Document Owner 252
Abbreviations 252
I.B.1 Overview 253
I.B.2 Application Overview 253
I.B.2.1 Key Contacts 253
I.B.3 Critical Information 253
I.B.3.1 Application Architecture 253
I.B.3.2 Application Dependencies 253
I.B.3.3 Backup Requirements 253
I.B.3.4 Installation Instructions 253
I.B.3.5 Monitoring Instructions 253
I.B.3.6 Security Requirements 254
I.B.3.7 Disaster Recovery and Business Continuity 254
I.B.3.8 Database Architecture 254
I.B.3.9 Network Architecture 254
I.B.3.10 SAN Storage Architecture 254

Revisions

Record of Past Modifications

Below is a record of all modifications previously made to this document.

Version	*Date*	*Name*	*Details*

Document Owner

The individual named below is the owner of this document. No modifications can be made to the document without going through this individual. All modifications

© 2010 by Taylor and Francis Group, LLC

made to the document will be recorded by the document owner in the "Record of Past Modifications" section above.

[Document Owner]	
Job Title:	
Company:	
Dept:	
Phone:	
Mail Drop:	
Location:	
E-mail:	

Operational Readiness Testing (ORT) Completion Approval

Your approval indicates that you have reviewed the completed tasks and agree that the application/environment is ready for implementation to the production environment.

Technical Operations Lead

Design & Engineering

Project Manager

Functional Manager

Completion Certification

After completing the assigned task, indicate your completion and certification that the area tested is ready for implementation to the production environment.

© 2010 by Taylor and Francis Group, LLC

	Task	*Sign-off*
☐	Section 3.1: Common Task	_____
☐	Section 3.2: Tools Acceptance	_____
☐	Section 3.3: Monitoring Tools Acceptance	_____
☐	Section 3.4: Server QA Checklist	_____
☐	Section 3.5: Backup Acceptance	_____
☐	Section 3.6: Metaframe Servers	_____
☐	Section 3.7: UNIX Acceptance	_____
☐	Section 3.8: SAN Storage	_____
☐	Section 3.9: Database Configuration	_____
☐	Section 3.10: CMDB Configuration	_____

Abbreviations

Acronyms	*Description*
CI	configuration item
CMDB	configuration management database
CM	change management
D&E	Design & Engineering
DBA	database administrator
DNS	domain name system
FTP	file transfer protocol
HBA	host bus adapter
IDS	integrated data store
IIS	internet information service
JDBC	Java database connectivity
LUN	local unit number
NER	New Environment Request (phase of the RLC)

© 2010 by Taylor and Francis Group, LLC

NTP	network time protocol
OLA	operational level agreement
OR	Operations Readiness (phase of the RLC)
ORT	Operational readiness testing
OS	operating system
QA	Quality Assurance (phase of the RLC)
RFC	request for change
RLC	release lifecycle
RM	release management
SAN	storage area network
SLA	service level agreement
SSL	secured socket layer
WWPN	world wide port name

I.1 Overview

Operational readiness testing (ORT) is the last step of an application or hardware's migration into production. ORT is the confidence-building process each application must undergo prior to going live into the production environment. During ORT, the application is locked down and may not be changed or modified. It is the time when the locked application is tested after its documentation has been reviewed.

Any application that is migrated without going through ORT is less likely to be migrated successfully and will cause more incidents, and will experience more frequent loss of service and degraded availability than applications that do not go through the process. Failure to complete ORT could potentially result in a loss of data and cause a significant loss of revenue. As a standard, no application or server is to be promoted to production without first passing ORT.

Production is defined as the environment or state where there is an expectation from the end user that a product/system/service is generated and that the product/system/service is sustained at the agreed-upon level.

Typically the ORT process requires forty hours to complete. This time and the associated costs must be accounted for in the schedule of the project. In order to make this process run efficiently, it is important that all project resources are responsive to Technical Operations inquires and requests for information during this process. Nonresponsiveness will cause delays in the process and extend the time to complete the ORT process.

© 2010 by Taylor and Francis Group, LLC

I.2 ORT Process

Before an application or server can be implemented into a production environment it must be tested to ensure its operational capacities and reliability. It must also be prepared for operation by ensuring that the appropriate system monitoring tools, backup procedures, security tools, and virus tools are installed, tested, and operational. The process also must ensure, that there is a disaster recovery and business continuity plan in place.

The process also ensures that the configuration management database (CMDB) is updated in support of sustaining current and relative information for post-ORT operational objectives.

I.2.1 ORT Preparation

Prior to entering the ORT process specific overview documentation and artifacts must be supplied by the project team to ensure a comprehensive understanding of the application or infrastructure and its contingencies. The documentation needs to be understood by:

- Technical Operations (Tech Ops)
- Network Operations (Net Ops)
- Configuration management
- Project managers
- Project teams and functional teams

I.2.1.1 Summary Artifact

The Key Application Information Guide (Attachment B) is used as the summary artifact that must be completed prior to entering the ORT process. It provides complete information about the application or infrastructure that is going to be prepared for implementation. The guide contains the following categories:

- High-level Application Description
- Primary/Secondary Contacts
- Details of Critical Information
- Logical/Physical Architecture Diagram (all environments)
- Interdependencies Context Diagram
- Backup Requirements
- Security Requirements
- Disaster Recovery and Business Continuity Plan
- Database Architecture
- Network Information
- Hardware/Software Requirements
- Installation Instructions
- Monitoring Schedule

© 2010 by Taylor and Francis Group, LLC

I.2.2 Preprocess

Prior to launching into ORT, Design & Engineering (D&E) must check the application's robustness. This process confirms such things as:

- Backups being made
- Backups being restored
- Security patches have been installed
- Local admin accounts have been deleted
- The CMDB has been updated
- Agents have been installed
- Operating System (OS) configuration parameters are properly configured

Tech Ops must be able to support the environment at the operating system level. During this period of time the application team must comply with the freeze period policy. During ORT, only the Tech Ops team will be able to touch the environment. The ORT process goes smoother and quicker if all artifacts are accurate.

I.2.3 ORT Steps

The ORT is initialed after approval is received via the change management process. The project manager (PM)/application team initiates the request for change (RFC) and is placed on the forward schedule of change (FSC). As part of the change process, the change owner sends an e-mail alert at the start of the ORT that kicks off the freeze period.

Tech Ops has the ability at the operating system level to lock out the application team members. The application team must refrain from touching the boxes during the ORT period.

The ORT steps are:

1. Documentation is completed; an RFC is initiated by the PM/application team and is submitted to change management for approval and scheduling.
2. Upon approval to begin ORT, change management issues several work orders; one for each of the nine primary ORT activity areas. The work orders are assigned to the operations process lead to begin the ORT process for the scheduled date. Work orders should be created as needed for:
 a. Common task
 b. Tools acceptance
 c. Monitoring tool acceptance
 d. Windows server QA checklist
 e. Net backup acceptance
 f. Metaframe servers
 g. UNIX acceptance

© 2010 by Taylor and Francis Group, LLC

h. Database configuration
i. CMDB configuration

3. The process lead reviews the work orders, the details of the RFC, and detailed information segments of the supplied system documents and the ORT form for:
 a. Names of all the associated servers
 b. Names of all the associated infrastructure
 c. Names of all the associated applications
 d. Names of all the associated services
 e. Others as needed

4. The process lead delegates the work orders to the appropriate technical resources to complete their assigned tasks within their technical specialty.
5. The technical staff validates the servers, tools, and OS (*not* the hardware or application).
 a. Validation is documented on the respective checklists in Section I.3 of this document.
 b. Approve that the tasks have been completed in the approval section of this document.
 c. Document activities in the detail box of the work order.
 d. Close assigned work orders.

It should be pointed out that if any of the ORT tasks fail testing, this will impact the completion of the ORT process and will affect the time to complete the process and exceed the operational level agreement (OLA) of forty hours. D&E may need to be reengaged for evaluation of the failure and subsequent configuration change may need to be made.

I.2.4 Updating the CMDB

Prior to entering into the ORT process, the subject Configuration Items (CIs) of the ORT must be listed in the configuration management database (CMDB); this should occur during the New Environment Request (NER) phase of the release lifecycle. The ORT process should not commence until this has been verified. There will be times when it is discovered that there are no records for the subject CIs in the CMDB. If this occurs, it is the responsibility of Design & Engineering and the project manager to ensure that records for the CIs are created within in the CMDB. This is done by working with configuration management.

Part of the ORT process is to ensure that the attributes of the new CIs have been updated within the CMDB. This is accomplished by completing the CMDB Configuration Checklist found in Section I.3.9 of this document. The checklist should be confirmed by configuration management to ensure proper updates have been completed.

© 2010 by Taylor and Francis Group, LLC

I.3 ORT Checklist

The ORT Checklist has been provided to ensure that all areas of the ORT have been reviewed and completed. The checklist should be completed by the resource completing the assigned task.

I.3.1 Common Task List

Common Tasks	*Status*
General Task	
Name of the server:	
Name of the database:	
IP address of the server:	
Proper VLAN:	
Operating system:	
Name of the application:	
Environment:	
Security	
Check that latest service pack installed	
Check that the latest IE is installed with latest service pack	
Check for any unwanted software other than that required	
Check that all the recommended patches are installed	
Check the existing user validation	
Check that no local admin account is present other than administrator	
Confirm the local admin password	
Check that no user exists with user ID 0 other than root (UNIX)	
Check that the server is hardened as per defined in the hardening document	

© 2010 by Taylor and Francis Group, LLC

I.3.2 Tools Acceptance Task List

Tools Tasks	Status
UNIX	
Audit ■ Action Manager Service ■ Audit Log Router Service ■ Audit Redirector Service	
Asset Management ■ Asset Management Agent Service	
Software Delivery ■	
Windows	
Audit ■ Action Manager Service ■ Audit Distribution Agent Service ■ Audit Log Router Service ■ Audit Portmapper Service ■ Audit Redirector Service ■ Gateway Service	
Virus Software ■ The product ENGINE and DAT is the latest ■ The upgrade and update is scheduled for everyday ■ The full scan is scheduled	
Asset Management ■ Asset Management Agent Service	
Software Delivery ■	

© 2010 by Taylor and Francis Group, LLC

I.3.3 Monitoring Tools Acceptance Task List

Item	Tasks	Status
	Windows	
1	Check if service is present and is running.	
2	Check agents.	
	<insert other tasks as needed>	
	UNIX	
1	Check operations agent.	
2	Check performance agent.	
	<insert other tasks as needed>	

I.3.4 Server QA Checklist for Windows Operations

QA Tasks	Pass/Fail	Verification Name	Date
Basic OS & Server			
Base OS and patches			
Server baseline hardening is done per the security doc			
Check IE version and service pack			
Serial number is recorded			
Run for security check			
Serial number is recorded			
Check Administrators, Power Users, and other Groups			
Test administrator password			

© 2010 by Taylor and Francis Group, LLC

QA Tasks	*Pass/Fail*	*Verification Name*	*Date*
Network			
Server cables fastened securely and network is ready			
Primary and failover network (teaming)			
Backup network			
IP address, domain name system (DNS), gateway, and other configuration			
Check primary heartbeat, failover heartbeat configuration, if applicable			
Terminal services configured			
Check for share folders			
Antivirus/Patch			
In Windows install/upgrade to the latest version			
Latest hotfix and patches need to be installed			
NetBackup			
NetBackup client installed/configured			
Error control coding Master Agent installed/configured			
<Insert activities for other tools>			

© 2010 by Taylor and Francis Group, LLC

QA Tasks	Pass/Fail	Verification Name	Date
Other			
Complete CMDB Configuration Item form			
Remove temporary user accounts			
Network			
Duplex and speed correct			
Configure port with appropriate description for troubleshooting			
Comments			
Tech Ops QA Review and Approval:			

I.3.5 Backup Acceptance Task List

Backup Tasks	Status
For WINDOWS servers:	
1. Verify client version is installed	
2. Verify master & media server name is configured in client	
3. Verify services are started	
For UNIX servers:	
1. Verify client version is installed	
2. Verify master & media server name is configured	
3. Verify the client hostname entry is on the master & media servers	

© 2010 by Taylor and Francis Group, LLC

I.3.6 Checklist for Metaframe Servers

Tasks	Status
Verify operating system is up to date	
Verify all programs/hotfixes are installed	
Verify metaframe services are started	
If the server has the role of presentation server, verify if applications are installed and published; list applications that are not yet complete for every server	
If the server has the role of Web Interface, verify if the following services are started: ■ IIS Admin Service ■ World Wide Web Publishing Service	

I.3.7 UNIX® Acceptance Task List

UNIX Tasks	Status	Other
DSView connection (terminal access) is configured and tested.		
Root file system should be on local disks and mirrored.		
Crash dump is enabled and Filesystem has enough space to save the dump.		
Swap space must be in accordance with physical memory (at least equal to RAM).		
Check backup interface (private interface).		
Check backup network routes.		
Check named is disabled.		
Check that network time protocol (NTP) is configured properly.		
Check that network cards are configured on full speed and full duplex.		
Sendmail should be disabled, unless required.		
Check time zone.		

© 2010 by Taylor and Francis Group, LLC

UNIX Tasks	Status	Other
Network parameters for interface cards are configured per security requirements.		
File permissions are set properly on system folders per security requirements.		
Root login is restricted to console only.		
Logging is enabled for file transfer protocol (FTP).		
Trust relationship between hosts is disabled by default, unless required.		
Check naming standards for accounts and groups.		
Server configuration document.		
Check the password expiration parameters.		
Check that the password expiration has been removed from service accounts.		
Check license expiration for the software and list of license keys.		
Check DNS entries and resolutions.		
Jumpstart document.		
Server build document for the environment/applications.		
Process to be followed for getting root password.		
Check that backup client configured and working.		
Admin account to access to the cluster.		
Cluster configuration testing.		
Check that accounts are locked for generic OS accounts.		
Check that only authorized users have UNIX accounts.		
Make sure log rotation is enabled for process accounting in the logadm.conf file (for UNIX Box).		

© 2010 by Taylor and Francis Group, LLC

I.3.8 SAN Storage Checklist

SAN Storage Tasks	Pass/Fail	Date
For UNIX Servers:		
Run "grab" script.		
For Windows Servers:		
Run "reporter" script.		
For All Servers:		
Verify that the server is using drivers and software on the most current SAN Standard Software List.		
Verify that the server is connected to two separate SAN switches.		
Verify that the SAN Patch Panel spreadsheet matches server name, switch port, and worldwide part name (WWPN).		
Verify that the server is properly zoned into appropriate storage device.		
Verify that the Volume Manger is installed and recognizes all devices presented to server.		
Validate connectivity with server connection to fabric A shutdown.		
Restore server connection to fabric A and verify both paths.		
Validate connectivity with server connection to fabric B shutdown.		
Restore server connection to fabric B and verify both paths.		
Verify that the server CI is updated in the CMDB host bus adapter (HBA), Storage System, and local unit number (LUN) info.		
For Clustered Servers:		
Verify that all LUNs on the server are presented to all cluster nodes.		
Verify that the SAN Clustering Worksheet is updated with LUN information.		

© 2010 by Taylor and Francis Group, LLC

I.3.9 Database Configuration Checklist

Database Tasks	Status
Planning and Risk Management	
Identify and patch known and reported vulnerabilities.	
Identify and record software versions and patch levels on the system. This is to be considered as a severity level 01 issue.	
Install only the database features that are needed.	
Record database configuration and store securely.	
Record database security configuration and store securely.	
Review database security procedures and policies.	
Store copies of the media used to build the database.	
Consider the physical location of servers.	
Define the database/application architecture.	
Host Operation System Security Issues	
Check for the following and maintain the relations.	
Check owner of software owns all files.	
Lock software owner account meets the standards.	
Limit access to software owner account.	
Use separate owners for different components.	
Check trace file permissions.	
Check permissions of the data files that only the owner should be able to read, write.	
Monitor log files.	
Check for sensitive temporary files.	
Check for tertiary trace files.	
Check for remote data access files.	
Raw device permissions.	

© 2010 by Taylor and Francis Group, LLC

Database Tasks	*Status*
Usernames and passwords in process list.	
Restrict the ps command.	
Search shell history files for usernames and passwords.	
Secure network transmissions.	
Encrypt data transmissions.	
Secure password transmission on the server.	
Secure password transmission on the client.	
Java database connectivity (JDBC) thin driver transmissions—ensure minimum permissions of connections are used.	
Audit environment variables for usernames and password.	
Audit the machine for scripts containing usernames and passwords.	
Audit chronology for usernames and passwords.	
Audit client machines for configuration files containing usernames and passwords.	
Remove database creation scripts.	
Utilize OS auditing facilities.	
Save log files to a separate server.	
Integrity check OS files.	
Consider using host based integrated data store (IDS).	
Review expected processes regularly.	
Check control file permissions.	
Confirm who is creating trace files.	
Audit trace files for attempts to read database internal structures.	
Ensure that no user has “Alter Session” and “Alter System” privileges.	
Audit for export file existence.	
Changing database passwords after full database import.	

© 2010 by Taylor and Francis Group, LLC

Database Tasks	*Status*
Locate archive log files and check that no user except the software owner can read them.	
Audit external tables used.	
If the servers are clustered servers and they are to support OS-level failovers within the clusters, check for the resources on the system because the server node is to support all the databases in a worst-case scenario.	
To meet such a contingency check resources like RAM and kernel parameters.	
Database and User Management	
Audit database users activities.	
Audit application database logins.	
Audit users database passwords.	
Establish a policy that prevents users from sharing account IDs.	
Audit default database accounts.	
Add password management for default accounts.	
Audit internal alias login.	
Change system password.	
Disable remote login password file.	
Check use of system tablespace as default.	
Audit known default role passwords.	
Audit users' accounts for passwords that are the same as the username.	
Audit users' accounts for weak passwords.	
Lock dormant database accounts and remove after time delay.	
Stop personal data exposure on users' accounts.	
Use obfuscated naming convention for users' accounts.	
Review database accounts, ensuring they belong to business users.	
Secure remote password login file.	

© 2010 by Taylor and Francis Group, LLC

Database Tasks	*Status*
Access Controls	
Audit third-party and homegrown applications authentication.	
Audit Java access to the OS.	
Secure all user views.	
Understand Data Access Descriptor administration.	
Secure access to catalog roles.	
Secure access to database administrator role views.	
Password protect admin roles.	
Check role hierarchy depth.	
Adopt role naming conventions.	
Create a role to manage users' accounts.	
Check for users who have dba privilege.	
Check for users or roles that have been granted "All Privileges."	
Check for privileges with keyword granted.	
Review system privileges granted.	
Check for application objects owned by privileged users.	
Check for direct access granted to tables and objects.	
Use Integrity constraints.	
Use triggers to insert critical data.	
Restrict users to one role at a time.	
Do not use remote host-based authentication.	
Audit public execute privileges on system-owned packages.	
Audit directly granted privileges.	
Access tables through packages or roles.	
Set up profiles for each class of database user.	
Set up general profile parameters.	
Review hidden initialization parameters.	

© 2010 by Taylor and Francis Group, LLC

Database Tasks	*Status*
Ensure system triggers fire.	
Objects in application tablespaces not owned by the schema owner should be dropped.	
Audit quota use per user.	
Establish different users for schema management and data management.	
Set up naming conventions for schema owners and administrators and users.	
Audit users' database triggers.	
Auditing	
Configure audit and storage.	
Audit insert failures on critical objects.	
Use triggers to capture login events.	
Audit create session.	
Audit use of all grant privileges.	
Audit the use of all drop statements.	
Audit the use of all alter statements.	
Audit the use of create user.	
Audit use of create role.	
Audit all create statements.	
Establish procedures to review audit logs.	
Use Log Miner to audit in the case of forensics.	
Configure basic audit.	
Limit users who can change the audit trail.	
Protect the audit trail.	
Backup the audit trail.	
Purge the audit trail.	

© 2010 by Taylor and Francis Group, LLC

Database Tasks	*Status*
Networking	
Prevent set commands on the listener.	
Force the MTS dispatcher to use specific ports.	
Use a personal firewall on database administrator computers.	
Restrict sources of database connections.	
Audit database links for hard cleartext passwords.	
Discover what objects can be seen in the linked database.	
Create a policy to manage database links.	
Audit what links exist into and from the database.	
Confirm the file permissions in the network admin directory.	
Add only minimum configuration files to all clients.	
Use an application gateway firewall.	
Availability/Backup/Recovery	
Enable secured socket layer (SSL) to protect client transmissions.	
Review and document backup and restore procedures.	
Review and document recovery procedures.	
Store backup media off site.	
Schedule cold backups.	
Validate the backup media regularly.	
Do not allow backups to be available online.	
Create and use media retrieval procedures.	
Mirror the online redo logs.	
Ensure the database is in archive log mode.	
Ensure archive log directories exist and are protected.	
Ensure archive logs are written to backup and are purged.	
Magnetically wipe old disks that have contained database data.	
Document and review disaster recovery procedures.	

© 2010 by Taylor and Francis Group, LLC

Database Tasks	*Status*
Include business users in disaster recovery planning.	
Application Development	
Precompile Java code before loading into the database.	
Review which applications access the database and how and from where.	
Implement procedures to limit the applications that can access the database and from where.	
Limit administration tools from accessing the database.	
When decommissioning old applications remove all binaries and files.	
Review procedures for adding new applications.	
Establish procedures for movers, leavers, and joiners.	
Audit application file permissions.	
Check for evidence of development on production databases.	
Restrict ad-hoc queries against the production database.	
Review users' permissions in test and development databases.	
Check for database links with access to production databases from development or test systems.	
Ensure the restored production data held in test or development has been deleted.	
Do not locate test and development databases on the same server as production.	
Ensure that there is no access from test and development to production.	
No developer access to production.	
No developer database accounts should exist on the production database.	
Ensure that backups and exports copy passwords to test and development are different.	
Place development and test on a different network segment than production.	

© 2010 by Taylor and Francis Group, LLC

Database Tasks	*Status*
Move all nonapplication objects from application tablespaces.	
Ensure no privileged user owns application objects.	
Audit the resources used by the database.	
Do not duplicate Oracle authentication.	
Do not use one database login to authenticate all other users.	
Do not use schema owners for administration tasks.	
Ensure the schema owner is not a dba.	
Lock schema owner accounts.	
Audit public synonyms.	
Do not hard code usernames and passwords in application source code.	
Consider not using Java.	
Do not allow applications to change the schema.	
Batch processes should access the database through one designated account.	
Do not use external accounts for batch processes.	
Consider password retrieval and use in schedulers.	
Enable batch database accounts only when needed.	
Audit query tool privileges.	
Encrypt critical data.	
Audit generated applications for known weaknesses.	
Audit public libraries used for known vulnerabilities.	
Use change control.	
Audit use of advance queues.	
Audit tools used for password leakage.	
Ensure no tool offers better access to the database than the application.	
Checksum application files for Trojans.	

© 2010 by Taylor and Francis Group, LLC

Database Tasks	Status
Restrict databases that can be accessed.	
Review how to enable and disable various database access features.	
Protect debugger interfaces.	
Do not divulge system information to the public.	

I.3.10 CMDB Configuration Checklist

CMDB Tasks	Status
Name	
OS	
Serial No.	
Product ID	
IP_Address	
Environment	
Application	
Clustered (Yes/No)	
Managed_By	
Monitored_by_HPOV (Yes/No/Unknown)	
Admin_Privilege	
User_Privilege	
Vendor_Managed	
Application_Group	
Location	

© 2010 by Taylor and Francis Group, LLC

Attachment A: Roles and Responsibilities

Operational Readiness Testing

	Project Mgt/ App Team	*D&E*	*Tech Ops*	*Net Ops*	*Change Mgt.*	*Config. Mgt.*	*Release Mgt.*
Initiate ORT process	A/R	C					C/I
Key Application Guide	A/R	R		R		C	I
Creation of RFC	A/R	C		C			
Change approval and scheduling	A	C	I		R	I	I
Creation of work orders	C	C/I	I	C/I	A/R		
Review of work orders, RFC, CMDB		C	A/R	C		C	
Create CI record in CMDB if not already created		A/R	C			R	
Assign work orders, create ORT Manual			A/R				
Respond to ORT questions	R	R	A/R	R		C	
Complete ORT tasks, and update CMDB			A/R			R	I

© 2010 by Taylor and Francis Group, LLC

	Project Mgt/ App Team	*D&E*	*Tech Ops*	*Net Ops*	*Change Mgt.*	*Config. Mgt.*	*Release Mgt.*
Sign off completion and route ORT Manual to release management	I	I	A/R	I			C/I
ORT Manual reviewed for completion	I	I	C	I			A/R
Environment turned over	I	I	A/R	I			I

A—Accountable (oversees task to ensure it is completed according to expected standard)

R—Responsible (performs action, assignment, task)

C—Consulted (provides additional information and helps determine, presence required in meetings)

I—Informed (provide with updates, no action required)

Attachment B: Key Application Information Guide*

Revisions

Record of Past Modifications

Below is a record of all modifications previously made to this document.

Version	*Date*	*Name*	*Details*

* [Application Name]
Business Technology Solutions
Key Application Information Guide
Version [x.x]

© 2010 by Taylor and Francis Group, LLC

Document Owner

The individual named below is the owner of this document. No modifications can be made to the document without going through this individual. All modifications made to the document will be recorded by the document owner in the "Record of Past Modifications" section above.

Name:	
Job Title:	
Company:	
Dept:	
Phone:	
Mail Drop:	
Location:	
E-mail:	

Abbreviations

Acronyms	*Description*
CI	configuration item
CMDB	configuration management database
CM	change management
COE	Center of Excellence
NER	New Environment Request
OR	Operations Readiness
ORT	operational readiness testing
QA	Quality Assurance
RFC	request for change
RLC	release lifecycle
RM	release management
SAN	storage area network

© 2010 by Taylor and Francis Group, LLC

I.B.1 Overview

The Key Application Information Guide is intended to be used to provide core/required application information necessary to successfully conduct operational readiness testing (ORT). It is also used to document key configuration information necessary to optimize the new environment being implemented.

I.B.2 Application Overview

[This section describes at a high-level the application, its function, key users, and associated criticality.]

I.B.2.1 Key Contacts

[This section describes the key application and support contacts and contact information.]

I.B.3 Critical Information

[This section describes the key information required, including the release schedule.]

I.B.3.1 Application Architecture

[This section describes and logically/physically diagrams the application architecture, integration points, database, and server environments.]

I.B.3.2 Application Dependencies

[This section describes the interdependencies and enterprise architecture integration points.]

I.B.3.3 Backup Requirements

[This section describes the system backup requirements, method, schedule, and process]

I.B.3.4 Installation Instructions

[This section describes installation instructions.]

I.B.3.5 Monitoring Instructions

[This section describes monitoring instructions including schedules.]

© 2010 by Taylor and Francis Group, LLC

I.B.3.6 Security Requirements

[This section describes the operating system security requirements.]

I.B.3.7 Disaster Recovery and Business Continuity

[This section describes the disaster recovery and business continuity plan for this application.]

I.B.3.8 Database Architecture

[This section describes database architecture, software, version, instance information, database server names, and relative database cluster information.]

I.B.3.9 Network Architecture

[This section describes the network requirements. This includes required firewall port information and access rules.]

TCP/UDP Port Numbers	Source	Source IP	Destination	Destination IP	Direction of Flow	Comments

SRL Number	Service Description	TCP Port	UDP Port	Comments

I.B.3.10 SAN Storage Architecture

[This section describes the SAN storage requirement, including SAN-connected servers and storage tiers.]

© 2010 by Taylor and Francis Group, LLC

Appendix J: Support and Escalation Turnover Document*

Contents

Revisions 256
Record of Past Modifications 256
Document Owner 256
Abbreviations 257
Support Completion Approval 257
Service Desk Certification 258
J.1 Overview 258
J.2 Completion of the Document 259
J.3 Support Preparation 259
J.3.1 Workgroup Creation 259
J.4 Service Level Definitions 259
J.5 Application Overview 260
J.6 Installation and Configuration 261
J.7 Implementation Support 262
J.7.1 Implementation Support Transition Plan 262
J.8 Steady-State Support 263
J.8.1 Support Contact Matrix 263
J.8.2 Escalation Paths 264
J.8.3 General Support Information 265

* [Name of Service]
Support and Escalation Procedures Service Desk Turnover
Version x.x

© 2010 by Taylor and Francis Group, LLC

J.8.4 Vendor/ASP Support Information....265
J.8.5 Call Logging....266
J.8.6 Known Errors....266
J.9 Escalation Procedures....266
Attachment A: Roles and Responsibilities....270

Revisions

Record of Past Modifications

Below is a record of all modifications previously made to this document.

Version	*Date*	*Name*	*Details*

Document Owner

The individual named below is the owner of this document. No modifications can be made to the document without going through this individual. All modifications made to the document will be recorded by the document owner in the "Record of Past Modifications" section above.

[Insert name of document owner]	
Job Title:	
Company:	
Dept:	
Phone:	
Mail Drop:	
Location:	
E-mail:	

© 2010 by Taylor and Francis Group, LLC

Abbreviations

Acronyms	*Description*
ASP	application service provider
OR	Operational Readiness
ORT	operational readiness testing
OSST	operational support service team
RLC	release lifecycle
RM	release management

Support Completion Approval

Your approval indicates that you have reviewed the support and escalation tasks and agree that the application/environment is ready for implementation to the production environment.

Operational Support

Incident Management

Service Desk

Release Management

Project Manager

Functional Manager

© 2010 by Taylor and Francis Group, LLC

Service Desk Certification

Your signature indicates your completion and certification that the tasks listed below have been completed and the subject application/system is ready for steady-state support on the date listed below. This section is to be completed by the service desk.

Date of Implementation:		*Date of Service Desk Turnover:*	

Acknowledgement of Receipt	Date
Support Document Review Meeting Completed	Date
Support Documentation Posted in Knowledge Base	Date
Customer Service Rep's Training Completed	Date
Deskside Training Completed	Date
Steady-State Support Ready	Date
Operational Support Acknowledgement	Date
Release Management	Date

J.1 Overview

The purpose of this document is twofold: (1) Provide an overview of the implementation support and escalation process that outlines the method of support and the role of the project team and the service desk. (2) Supply the service desk with the necessary information to be able to resolve service outages in the most expeditious manner and restore service to the agreed-upon service level. It is meant to increase the availability, reliability, and maintainability of a new system or service.

The document will clearly describe and outline the incident escalation flow and document any known errors and workarounds. The type of expected service

© 2010 by Taylor and Francis Group, LLC

request associated with this application will also be documented. The document covers areas such as:

- Installation and configuration
- Application owners
- Escalation procedures
- Known errors and workarounds
- Common user errors

J.2 Completion of the Document

An important aspect of providing a high level of support is proper documentation and timely implementation of the support processes; therefore it is extremely important that the turnover document be completed accurately and executed thirty (30) working days prior to implementation. This allows for training of service desk resources, app support resources, documenting knowledge bases, and verification of understanding of the new support process.

Completion of this document should be coordinated through the release project manager in conjunction with the project team. A review of the document must be done with the operational support services team (OSST) to ensure understanding and completeness. The OSST will be the liaison with the service desk and ensure implementation.

J.3 Support Preparation

Prior to implementation, there is a need to prepare for user support of the application. The sections below provide a template for many of these items.

J.3.1 Workgroup Creation

A key item that needs to be created prior to implementation is the workgroup name or resolver group. This can be done by contacting the service desk and requesting that a workgroup and a resolver group be created.

J.4 Service Level Definitions

Service levels are created to establish prioritization of incidents and service requests based on urgency and impact to the organization. An incident may have a low impact but have urgency based on the number of users affected. Conversely, an incident may have a high impact and low urgency based on the application and time of day. Service levels are also a way to establish performance levels based on severity. Service level definitions are listed in the table below.

© 2010 by Taylor and Francis Group, LLC

Severity Level	*Description*
Severity 1	■ Loss of service resulting in a severe business impact or significant performance degradation ■ Problem with any system that has a significant effect on the loss of revenue, brand image, or financial impact ■ The number of impacted end users is significant or end user's function is business critical ■ No known workaround is available
Severity 2	■ Loss of service resulting in moderate business impact or performance degradation ■ Incident could have a financial impact or effect on the brand if not mitigated ■ The number of impacted end users is not as significant as a severity 1 incident ■ There is a known workaround that can be utilized
Severity 3	■ Loss of service to an individual end user with minor business impact ■ Does not have a time criticality
Severity 4	■ Has little effect if any on system availability ■ Primarily informational

J.5 Application Overview

Application Name	
Acronym	
Application Owner	Phone #
Business Owner	Phone #
What does the application do? What is it used for?	
Intended Users	
Application Name	

© 2010 by Taylor and Francis Group, LLC

Is this application considered a "critical" application *(a critical application is defined as an application that has a significant financial, brand, or revenue loss impact or prohibits a significant number of resources from being able to complete their job task if the service is not available for a short period of time)*? If so, why?	
Other Systems Impacted	
Is this application in a critical revenue path? If so, how?	
Is the app an application service provider (ASP)?	
Who is the vendor?	
Who owns the contractual relationship with the vendor?	

J.6 Installation and Configuration

Application Environment	
Installation:	
Standard desktop image?	
Workgroup that performs installation	
How to request access/account? Other? Explain	
SEC1: ☐ Yes ☐ No UIDRS: ☐ Yes ☐ No	
Password required?	
User ID format	
Application Environment	
Platform:	

© 2010 by Taylor and Francis Group, LLC

(Mainframe, UNIX®, shrink wrap app, database, etc.)	
Server name	
Server location	
Web Site URL:	
Who supports the Web site?	

J.7 Implementation Support

As with any new application or service, there will be an implementation period where the project team will closely monitor the application's performance and be available to answer user questions regarding the functionality. The project team will also be readily available to handle any incidents or service disruptions that occur. This implementation period can be handled in many ways. It is important that the service desk and OSST understand this implementation plan.

J.7.1 Implementation Support Transition Plan

Date of Implementation:	*Date of Turnover to Service Desk:*
Resolver Workgroup Name(s):	
Describe initial implementation support plan: (Type—war room, virtual staffing, phone numbers, hours of operation, what kind of support will be offered, etc.)	
Implementation support type: ☐ war room ☐ virtual (dedicated phone support by project team) ☐ service desk ☐ other (describe)	
Staffed support hours (please identify if hours differ during after-hours periods):	
Describe the type of support provided (full support, user support, password resets, etc.):	
Describe escalation process during implementation support period (please include service desk responsibilities during this period):	

© 2010 by Taylor and Francis Group, LLC

Date of Implementation:		*Date of Turnover to Service Desk:*
Describe the transition from implementation support to steady-state support by the service desk. Please provide timelines, length of implementation support, transition training, and documentation:		
Implementation Support Contact Information		
Role	*Contact Name*	*Contact Information*
Implementation Support Desk		
Functional Lead		
Project Lead		
Project Manager		
Problem Coordinator		
[Other role]		
[Other role]		

J.8 Steady-State Support

This section describes essential information needed to provide the day-to-day steady-state support of the application. Please complete all sections completely as this will be used by the service desk to support your application. Please attach any system diagrams or other supporting documents. It should be noted that the date steady-state support begins is in the "Date of Turnover" listed in Section 7.1.

J.8.1 Support Contact Matrix

Please provide contact information including after-hours contact information. Please insert additional fields as needed to add more contact information if needed. This support matrix will be utilized by the service.

© 2010 by Taylor and Francis Group, LLC

Role	*Contact Name*	*Contact Phone Numbers*	*E-Mail*
Service desk			
Level 2 support			
Level 3 primary			
Level 3 secondary			
Functional Lead			
Functional Manager			
Senior Manager			

J.8.2 Escalation Paths

Severity 1 and 2	
Severity 1 and 2, types of issues:	[Describe in general terms the type of severity 1 and 2 issues; refer to Section 4 definitions.]
Escalation path during **Business** hours:	
Special Instructions:	
Escalation path **After Hours**:	
Severity 3	
Severity 3, types of issues:	[Describe in general terms the type of severity 3 issues; refer to Section 5 definitions.]
General escalation instructions:	
Communications	
Identify all of the e-mail groups that will need to be notified of a service interruption. If there are different groups for different situations, please identify those situations:	

© 2010 by Taylor and Francis Group, LLC

J.8.3 General Support Information

Application workgoup name	
Send screenshots to:	
Workgroup e-mail (please include the Internet address):	
Server workgroup name (if applicable):	
Hours of operation:	
Contact info (name & phone):	
Daytime:	
After hours:	

J.8.4 Vendor/ASP Support Information

Vendor/ASP name and support phone number as marked	
Vendor business hours:	
Support phone number during **business** hours:	[Please provide vendor's business hours.]
Support phone number **after hours**:	[Please provide vendor's after hours.]
What types of issues should the vendor service desk be contacted about?	
What are the negotiated service levels and service availability hours?	
What is the agreed-upon maintenance window?	
What type of information is needed when the vendor is engaged (e.g., account number, user ID, screen shots, etc.)?	

© 2010 by Taylor and Francis Group, LLC

J.8.5 Call Logging

Please describe any specific ticket detail that is required to be obtained by the service desk from the customer. This could include server name, screen shots, and so on. Identify any information that may be different if required by the ASP.

J.8.6 Known Errors

Please list any known errors and workarounds that have been discovered during development or implementation support that can be used by the service desk to increase first call resolution and minimize service disruption.

Known Error Number	*Type of Known Error*	*Workaround*

J.9 Escalation Procedures

This section documents detailed escalation procedures to resolve any issues that were discovered in implementation support or any anticipated support processes. These anticipated support processes can include unable to connect, data is not available/inaccurate, or resetting of passwords. Please be descriptive and use terms and key phrases that the users would use in describing an incident. In the escalation procedures, please be direct when describing how to resolve the incident. Provide troubleshooting scripts that will be used in the service desk knowledgebase. Please feel free to attach any support documentation that could be used to resolve incidents; for example, system diagrams, process flows, and so on.

[Application or Server Name]	
Intended for: ☐ Service Desk ☐ Deskside ☐ App Support ☐ 2nd, 3rd Level Support	
Software/Hardware Issue:	
Problem Description: ☐ Known Error	

© 2010 by Taylor and Francis Group, LLC

[Application or Server Name]	
Ticket Detail Requirements:	
Troubleshooting Steps:	
Resolution:	
If Resolution doesn't work, what information needs to be captured in the ticket before it is transferred?	
Which workgroup should it be sent to?	
How do you want this information SCIM'd (system/component/item/module)?	

[Application or Server Name]	
Intended for: ☐ Service Desk ☐ Deskside ☐ App Support ☐ 2nd, 3rd Level Support	
Software/Hardware Issue:	
Problem Description: ☐ Known Error	
Ticket Detail Requirements:	
Troubleshooting Steps:	
Resolution:	
If Resolution doesn't work, what information needs to be captured in the ticket before it is transferred?	
Which workgroup should it be sent to?	
How do you want this information SCIM'd (system/component/item/module)?	
[Application or Server Name]	

© 2010 by Taylor and Francis Group, LLC

Intended for: ☐ Service Desk ☐ Deskside ☐ App Support ☐ 2nd, 3rd Level Support	
Software/Hardware Issue:	
Problem Description: ☐ Known Error	
Ticket Detail Requirements:	
Troubleshooting Steps:	
Resolution:	
If Resolution doesn't work, what information needs to be captured in the ticket before it is transferred?	
Which workgroup should it be sent to?	
How do you want this information SCIM'd (system/component/item/module)?	

[Application or Server Name]	
Intended for: ☐ Service Desk ☐ Deskside ☐ App Support ☐ 2nd, 3rd Level Support	
Software/Hardware Issue:	
Problem Description: ☐ Known Error	
Ticket Detail Requirements:	
Troubleshooting Steps:	
Resolution:	
If Resolution doesn't work, what information needs to be captured in the ticket before it is transferred?	
Which workgroup should it be sent to?	
How do you want this information SCIM'd (system/component/item/module)?	

© 2010 by Taylor and Francis Group, LLC

[Application or Server Name]	
Intended for: ☐ Service Desk ☐ Deskside ☐ App Support ☐ 2nd, 3rd Level Support	
Software/Hardware Issue:	
Problem Description: ☐ Known Error	
Ticket Detail Requirements:	
Troubleshooting Steps:	
Resolution:	
If Resolution doesn't work, what information needs to be captured in the ticket before it is transferred?	
Which workgroup should it be sent to?	
How do you want this information SCIM'd (system/component/item/module)?	

[Application or Server Name]	
Intended for: ☐ Service Desk ☐ Deskside ☐ App Support ☐ 2nd, 3rd Level Support	
Software/Hardware Issue:	
Problem Description: ☐ Known Error	
Ticket Detail Requirements:	
Troubleshooting Steps:	
Resolution:	
If Resolution doesn't work, what information needs to be captured in the ticket before it is transferred?	
Which workgroup should it be sent to?	
How do you want this information SCIM'd (system/component/item/module)?	

© 2010 by Taylor and Francis Group, LLC

Attachment A: Roles and Responsibilities

	Project Team	*OSST*	*Service Desk*	*Application Support*	*Incident Mgt.*	*Release Mgt.*
Create Support and Escalation Document	A/R	C			C	I
Review implementation support	A	R		C	C	
Review steady-state support	A	R		C	C	
Pass steady-state to service desk	I	A/R	C			
Implement steady state	C	C	A/R	C	I	I
Ensure steady state is implemented	C	R	A		I	I
Implementation support process activated	A/R	I	I	I	I	I
Transition to steady state	R	A/R	R	C	I	
QA steady state		R	A/R			

Note: R – Responsible (performs action, assignment, task), A – Accountable (oversees task to ensure it is completed according to expected standard), C – Consulted (provides additional information and helps determine, presence required in meetings), I – Informed (provide with updates, no action required)

© 2010 by Taylor and Francis Group, LLC

Appendix K: Master Training Plan*

Contents

Training Plan Acceptance 272
Training Plan Completion 272
Revisions 272
Record of Past Modifications 272
Document Owner 273
Abbreviations (Please list any relevant abbreviations.) 273
K.1 Overview 274
K.2 General Training Information 274
K.2.1 Purpose and Scope 274
K.3 General Planning and Approach 274
K.3.1 Delivery Method 275
K.3.2 Training Materials 275
K.3.3 Project References 275
K.3.4 Roles and Responsibilities 275
K.4 Delivery 275
K.4.1 Curriculum and Learning Objectives 275
K.4.2 Training Techniques 275
K.4.3 Schedule 276
K.5 Training Evaluation 276
K.5.1 Metrics 276
K.5.2 Learning Objectives 276

* *[Name of release]*
Master Training Plan
Version x.x

© 2010 by Taylor and Francis Group, LLC

Training Plan Acceptance

Your approval indicates that you have read, understand, and agree with the training plan outlined in this document.

Signature **Date**

Business Stakeholder

Project Management

Project Manager/Release Project Manager

Release Management

Training Plan Completion

After completion of the training outlined in this document, you indicate that the training plan in this document has been completed.

Project Management

Project Manager/Release Project Manager

Revisions

Record of Past Modifications

Below is a record of all modifications previously made to this document.

Version	*Date*	*Name*	*Details*

© 2010 by Taylor and Francis Group, LLC

Document Owner

The individual named below is the owner of this document. No modifications can be made to the document without going through this individual. All modifications made to the document will be recorded by the document owner in the "Record of Past Modifications" section above.

[Name of document owner]	
Job Title:	
Company:	
Dept:	
Phone:	
Mail Drop:	
Location:	
E-mail:	

Abbreviations (Please list any relevant abbreviations.)

Acronyms	*Description*

© 2010 by Taylor and Francis Group, LLC

K.1 Overview

The objective of this training plan is to identify the training needs and approach planned for application users and support staff. The needs of these two groups will differ and consideration must be given to how these needs must be fulfilled.

Different systems require different training—drop-ins versus full-day formal training, self-paced tutorials versus hands-on classroom training. The rule of thumb for planning is that it takes three times for the normal mind to assimilate new information.

K.2 General Training Information

K.2.1 Purpose and Scope

Please describe the purpose and scope of the training plan. Include the resources that will need to receive training on the functionality and the support of the system/application. This may include end users, support resources, and superusers.

K.3 General Planning and Approach

Different approaches to training may be needed for users with varying organizational roles and varying abilities. Since different training "tracks" will likely follow different timelines and target audiences, planning the training curriculum and arranging for training facilities should begin weeks in advance of the actual training.

Schedule enough training time to ensure resources are prepared, but avoid scheduling training too far in advance of installation so that staff members don't forget what they have learned in the interim.

Training users too far ahead of installation means they'll have the opportunity to forget a fair amount of what they learned in class by the time the system is deployed. Also, try to avoid scheduling training classes during particularly busy times for departments (e.g., the close of the fiscal year, new installations, etc.).

When your objectives are clear, it's time to work on the details. Plan the sequence of each topic to be covered in the training class, determine the pace, and prepare materials accordingly, including an instructor's guide.

© 2010 by Taylor and Francis Group, LLC

K.3.1 Delivery Method

Please describe how the training for each group will be delivered; include type (classroom, online, etc.), tools, location, trainer, and length of training. Include any prerequisites or knowledge needed for each type of training.

K.3.2 Training Materials

Please provide details of the types of training materials that will be used during the training. This should include such items as system functionality, resetting of passwords, support scripts, and security. Will the training materials be available after the initial training? Be sure to provide any updated or enhanced user guides and maintenance guides.

K.3.3 Project References

Please provide a list of references and documents that can be used to develop training (e.g., system flow diagrams, security documents, system guides, application documentation). Please list points of contact that can be referenced for information, troubleshooting, and user and maintenance guide updates, etc.

K.3.4 Roles and Responsibilities

Please identify the resources and their responsibilities in content development, training materials, and delivery of the training.

K.4 Delivery

K.4.1 Curriculum and Learning Objectives

Briefly describe the curriculum and training objectives for each type of training class (e.g., general user, superuser, service desk, support team). How will these objectives be measured?

K.4.2 Training Techniques

Describe the training techniques that will be utilized (e.g., instructor-led training, computer-based instruction, self-paced manual, hands-on training, etc.).

© 2010 by Taylor and Francis Group, LLC

K.4.3 Schedule

Please provide a training schedule that will include training dates, locations, instructors, and groups to be trained (specific lists of students can be kept in a separate document).

K.5 Training Evaluation

K.5.1 Metrics

Please provide key metrics that evaluate the actual number and type of resources that received training.

Groups Trained (include number):
Actual Duration of Training Sessions:
Number of Resources that Missed Training Sessions:

K.5.2 Learning Objectives

Please list the learning objectives established in Section 4.1 and provide feedback on whether the objectives were met.

© 2010 by Taylor and Francis Group, LLC

Appendix L: Service Level Agreement*

Contents

Revisions 278
Record of Past Modifications 278
Document Owner 278
Abbreviations 278
Service Level Agreement 279
L.1 Overview 280
L.2 Service Level Policy 280
L.2.1 Service Level Definitions 280
L.3 Service Description 281
L.4 Service Hours and Availability 281
L.5 Service Level Objectives 281
L.5.1 Acknowledgment and Communications 282
L.6 Underpinning Contracts 282
L.7 Major Incident Process 283
Attachment A: System Availability Calculations 283

* [Name of Service]
Service Level Agreement
Version x.x

© 2010 by Taylor and Francis Group, LLC

Revisions

Record of Past Modifications

Below is a record of all modifications previously made to this document.

Version	*Date*	*Name*	*Details*

Document Owner

The individual named below is the owner of this document. No modifications can be made to the document without going through this individual. All modifications made to the document will be recorded by the document owner in the "Record of Past Modifications" section above.

[Name of document owner]	
Job Title:	
Company:	
Dept:	
Phone:	
Mail Drop:	
Location:	
E-mail:	

Abbreviations

Acronyms	*Description*
ASP	application service provider
ETR	estimated time to recovery
OLA	operational level agreement
OR	Operational Readiness
RLC	release lifecycle

© 2010 by Taylor and Francis Group, LLC

Acronyms	*Description*
RM	release management
SEP/SDTO	support & escalation procedure/Service Desk Turnover document
SLA	service level agreement
SLP	Service Level Policy
UPC	underpinning contract

Service Level Agreement

I have read and agree to the level of service described in the attached document. This agreement remains valid until suspended by a revised agreement mutually endorsed by the signatories below. The agreement will be reviewed periodically for revisions. Minor changes may be recorded as an addendum to this agreement, providing they are mutually endorsed by the signatories.

Project Sponsor

Business Owner

Functional Manager

Application Senior Manager

Technical Operations Manager

Network Operations Manager

Operations Support Services Manager

Service Level Management

Contracts & Sourcing Manager (if vendor)

© 2010 by Taylor and Francis Group, LLC

L.1 Overview

The purpose of this document is to provide an understanding of standard service level-related definitions and standards that are set forth in the Service Level Policy. This document will also clearly document the agreed-upon service availability, maintenance windows, and service restoration targets that differ from the standard policy.

Service levels set forth in this document must be driven by the needs of the business and ensure that they allow for the business to meet or exceed its stated objectives. Consultation with the business is needed to ensure that there is an understanding of what is expected, what capabilities are available, and what the cost is of delivering those levels.

The contents of this document supplements the support and escalation procedure/Service Desk Turnover document (SEP/SDTO) that is utilized to define the support process of the application.

L.2 Service Level Policy

The Service Level Policy (SLP) defines standard agreed-upon service levels and service availability for applications within the environment. The policy defines the various types of key service targets that include not only restoration of service during an interruption of service, but also the availability of the application and expected maintenance periods. The SLP sets baseline service levels for any ASP or outside service provider through underpinning contracts (UPC). The SLP also sets forth standard severity level definitions.

L.2.1 Service Level Definitions

Service levels are created to establish prioritization of incidents and service requests based on urgency and impact on the organization. Service level definitions are listed in the table below.

Severity 1	■ Loss of service resulting in a severe business impact or significant performance degradation
	■ Problem with any system that has a significant effect on the loss of revenue, brand image, or financial impact
	■ The number of impacted end users is significant or end user's function is business critical
	■ No known workaround is available

© 2010 by Taylor and Francis Group, LLC

Severity 2	■ Loss of service resulting in moderate business impact or performance degradation
	■ Incident could have a financial impact or effect on the brand if not mitigated
	■ The number of impacted end users is not as significant as a severity 1 incident
	■ There is a known workaround that can be utilized
Severity 3	■ Loss of service to an individual end user with minor business impact
	■ Does not have a time criticality
Severity 4	■ Has little effect if any on system availability
	■ Primarily informational

L.3 Service Description

Please describe the application or service that is the subject of the service level agreement (SLA). The description needs to include business functions, deliverables, and all relevant information to describe the service and its scale, impact, and priority for the business.

L.4 Service Hours and Availability

Please provide the hours that the customer expects the application or service to be available. Please see Attachment A for definition of availability times.

	Hours of Operations *(Pacific Time)*	*% Availability*
Monday–Friday	(Please use military time.)	
Saturday–Sunday		
Month End		
Year End		
Holidays (list any specific holidays)		
Agreed-upon maintenance times		

L.5 Service Level Objectives

There are several service performance objectives for response to service interruptions. This section lists service level objectives for these various areas.

© 2010 by Taylor and Francis Group, LLC

L.5.1 Acknowledgment and Communications

Severity	*Workgroup time to Acknowledgment*	*Communications*
1	10 min	■ 10 min acceptance ■ 15 min communication sent ■ Status sent every hour or – Stated interval – Until resolved – Until downgraded to severity 3
2	10 min	■ 10 min acceptance ■ 15 min communication sent ■ Status sent every hour or – Stated interval – Until resolved – Until downgraded to severity 3
3	30 min	■ 30 min acceptance ■ 1–3 hours if directed by workgroup ■ Status sent as requested
4	60 min	■ 30 min acceptance ■ 1–3 hours if directed by workgroup ■ Status sent as requested

L.6 Underpinning Contracts

If a vendor or ASP is involved in providing the subject service, please provide negotiated service levels.

Severity	*Time to Acknowledgment*	*Time to Resolution*	*Time to Status Update*
1			
2			
3			
4			

© 2010 by Taylor and Francis Group, LLC

L.7 Major Incident Process

A major incident process has been developed and can be reviewed in the major incident procedure document. A major incident is defined as:

- An incident that impacts a business-critical system (STARS, GENESYS, etc.) during business hours and,
 - Is already classified as a severity 1 incident and there is no estimated time to recovery (ETR) after 30 minutes of investigation or the estimated time to resolution is more than 2 hours.
- An incident in which the entire site is nonoperative or enterprisewide service is unavailable, and the condition is causing a significant loss of revenue, degradation of customer service, or brand erosion.
- An incident in which multiple site or system impacts require coordinated recovery and communication.

Attachment A: System Availability Calculations

Hours Per Day	*Days Per Week*	*Downtime per Week (minutes)*				
		Service Level				
		95%	*98%*	*99%*	*99.99%*	*99.999%*
8	5	120	48	24	14	1.44
12	5	180	72	36	22	2.16
18	5	270	108	54	32	2.24
24	5	360	144	72	43	4.32
24	6	432	173	86	52	5.184
24	7	504	202	101	60	6.048

© 2010 by Taylor and Francis Group, LLC

- Cost to provide increased availability increases exponentially as the level of availability increases.
- These times are provided for comparison purposes only.
- Times represented are aggregated times and may not occur in one single instance.

© 2010 by Taylor and Francis Group, LLC

Appendix M: IT Service Continuity and Recovery Plan*

Contents

M.1 Service Continuity Plan Acknowledgment and Approval 287
M.2 Service Continuity Plan Receipt and Storage 288
M.3 Revisions 289
M.3.1 Record of Modifications 289
M.3.2 Document Owner 289
M.4 Abbreviations 290
M.5 Overview 290
M.6 Recovery Strategy Overview 291
M.6.1 High-Level Service Overview 292
M.7 <Application> Disaster Recovery Requirements 292
M.7.1 <Application> Recovery Objectives 292
M.7.1.1 Recovery Point Objectives 292
M.7.1.2 Recovery Time Objectives 293
M.7.1.3 Describe the Architecture That Supports the RPO and RTO 293

* [Application Name]
Business Technology Solutions
IT Service Continuity/Service Restoration Plan
Version (version of specific plan)
Revision date: mm/dd/yyyy
Next plan review date: mm/dd/yyyy
(6 months from last revision date)

© 2010 by Taylor and Francis Group, LLC

M.7.2 Hardware Requirements 293
M.7.2.1 Hardware Configuration 293
M.7.2.2 Software Requirements and Vendor Support 294
M.7.2.3 Special Software Configuration Instructions 294
M.7.3 Business-Critical Data 294
M.7.3.1 Workaround If Input Files Are Not Available 295
M.7.3.2 <Application> Impact 295
M.8 Recovery Information 296
M.8.1 Redundancies 296
M.8.2 Dependencies 296
M.9 Recovery Execution 296
M.9.1 Risk Assessment Worksheet 296
M.9.2 Recovery Procedures 297
M.9.2.1 Application Recovery Script Process for Day 1 Recovery 297
M.9.2.2 Application Recovery Script Process for Full Recovery 297
M.9.2.3 Requirements for Initial Batch Processing Prior to Restoration 298
M.9.2.4 Requirements for Initial Batch Processing after Restoration 298
M.9.2.5 Requirements for Normal Batch Processing after Restoration 298
M.10 Support Notification List and Responsibilities 298
M.10.1 Management Notification 298
M.10.2 IT Technical Support Notification 299
M.10.3 Vendor Key Contacts 300
M.10.4 Users Required for Recovery 300
M.10.5 Recovery Checklist Validation 301
Attachment A: Roles and Responsibilities 301
Attachment B: IT Recovery Planning Deliverables Checklist 302
Attachment C: Application Architecture Diagram 302
Attachment D: Definitions 302

© 2010 by Taylor and Francis Group, LLC

M.1 Service Continuity Plan Acknowledgment and Approval

Your approval indicates that you have read, understand and agree with the attached continuity and recovery plan outlined in this document.

Signature	**Date**
Business Stakeholder	
Functional Manager	
Data Center Operations	
Network Operations	
Storage Operations	
IT Service Continuity	
Release Management (not needed if biannual review)	
Other as needed	
Other as needed	

© 2010 by Taylor and Francis Group, LLC

M.2 Service Continuity Plan Receipt and Storage

I certify, by signature, that I have read, understand and agree with the attached continuity and recovery plan outlined in this document, and will appropriately store this document in such locations (i.e., home, car, office) for immediate access in support of a disaster recovery effort.

Signature	**Date**
Functional Manager	
IT Service Continuity Management	
Data Center Operations	
Network Operations	
Storage Operations	
Service Desk	
Incident Management	
Other as needed	

© 2010 by Taylor and Francis Group, LLC

M.3 Revisions

M.3.1 Record of Modifications

Below is a record of all modifications previously made to this document.

Version	*Date*	*Name*	*Details*

M.3.2 Document Owner

The individual named below is the owner of this document. No modifications can be made to the document without approval of the document owner. All modifications made to the document will be recorded by the document owner in the "Record of Modifications" section above.

[Functional manager]	
Job Title:	
Company:	
Dept:	
Phone:	
Mail Drop:	
Location:	
E-mail:	

© 2010 by Taylor and Francis Group, LLC

M.4 Abbreviations

Acronyms	*Description*
ASP	application service provider
BCM	business continuity management
BCP	business continuity plan
BIA	business impact assessment/analysis
CI	configuration item
CM	crisis management
CMP	crisis management plan
CMT	crisis management team
DR	disaster recovery
DRP	disaster recovery plan
ITSC	IT system continuity
ITSCM	information technology system continuity management
RA	risk assessment/analysis
RPO	recovery point objective
RTO	recovery time objective
SLA	service level agreement
UPS	uninterruptible power supply

M.5 Overview

Information technology service continuity management (ITSCM) is an important part of supporting business-critical processes. Dependency on IT services has become a core component of business services and continued availability is critical to ‘profitability and survival. These dependencies have necessitated the inclusion of IT contingency planning into individual business continuity plans.

As part of the enterprisewide crisis management plan (CMP), it is paramount to consider application/system continuity and recovery during the release design and build process. The documentation for disaster recovery (DR) and system continuity will be incorporated into the CMP.

© 2010 by Taylor and Francis Group, LLC

It is important to remember that IT system continuity (ITSC) and disaster recovery planning are ongoing, repeatable activities utilizing the disaster recovery planning life cycle. The disaster recovery documents are living items and must be maintained in an ongoing process.

This document details the instructions and procedures that are required to be followed to recover or continue the operation of application/systems and to maintain service continuity to the level defined or agreed upon with the business.

This document represents a detailed application-level disaster recovery plan that fully outlines and describes, on a step-by-step basis, exactly how to recover the application from a worst case (i.e., total loss of data center) scenario. The plan is written in precise detail so that anyone with the appropriate skills can perform the specific recovery operation. It is intended for use as documentation to aid the restoration of this application during a disaster.

Attachment B, the IT Recovery Planning Deliverables Checklist, is provided to ensure that key information is documented and detailed in the plan.

M.6 Recovery Strategy Overview

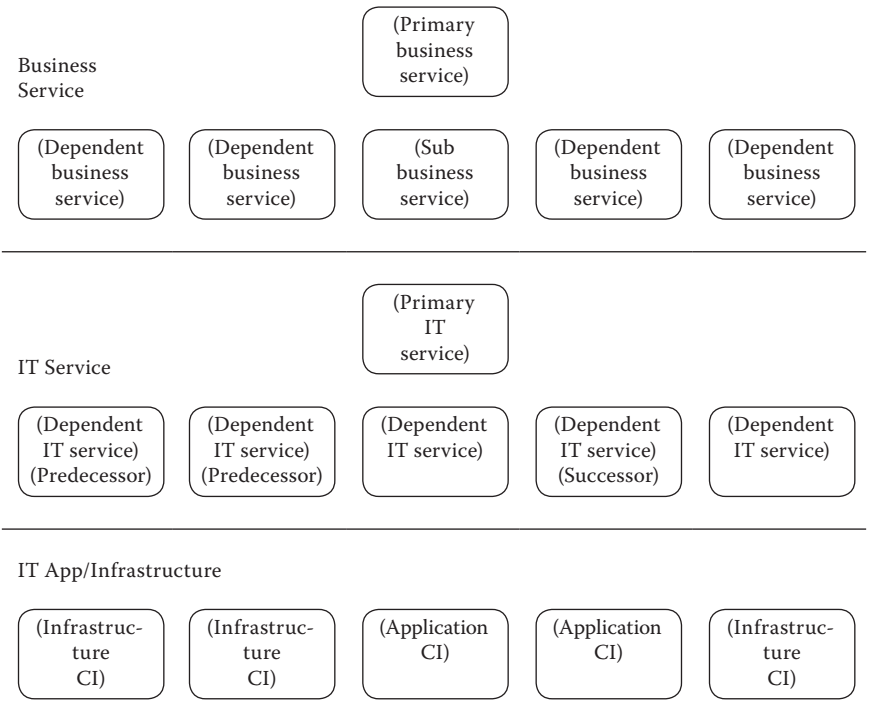
Please provide key recovery strategies associated with recovery of the subject IT service.

Affected business service and unit		
Primary contact (name, primary contact phone number, e-mail address)		
Type of system architecture		
In the event of an unplanned shut down, how long will it take to recover (recovery time objective, or RTO) the system, infrastructure, or service?		
What is the point (last commit) of data recovery (recovery point objective, or RPO)? (e.g., last known recovery point, data integrity, day/date/time)		
Have the procedures been tested?	Yes ☐	No ☐
Is there a vendor or application service provider (ASP) that should be considered in recovery efforts?	Yes ☐	No ☐
If *yes*, please provide the vendor name, address, contact name, and telephone number:		
Could the vendor or ASP be used as a recovery site?	Yes ☐	No ☐
Does vendor have documented disaster recovery plans?	Yes ☐	No ☐

© 2010 by Taylor and Francis Group, LLC

M.6.1 High-Level Service Overview

This section illustrates the top-down, bottom-up service relationship; connect items as necessary. For complete and comprehensive detail of this overview, please reference Attachment C, the Application Architecture Diagram.



M.7 <Application> Disaster Recovery Requirements

M.7.1 <Application> Recovery Objectives

This section describes the recovery objectives of the <Application>. Using the guidelines in the table below, determine how the application has been architected and built to determine the actual obtainable objective. The objective will commence *after* the prerequisite IT components—applications, hardware, and network infrastructures, etc.—have been restored and made available for service restoration to begin.

M.7.1.1 Recovery Point Objectives

Instructions: The recovery point objectives statement defines how much work in progress can be lost after a significant business disruption (for example, from point

© 2010 by Taylor and Francis Group, LLC

of impact, the previous backup, one week, etc.) Objectives must be based on current configuration and validated with testing.

The recovery point objective (RPO) for <Application> is _____.

M.7.1.2 Recovery Time Objectives

Instructions: This section describes the requirements for the recovery time objectives. Document how long the application can be out of service before the function becomes critical to daily business operations. List how soon the application must be recovered from point of impact. Legal and regulatory requirements must be considered.

The recovery time objective (RTO) for <Application> is _____.

M.7.1.3 Describe the Architecture That Supports the RPO and RTO

Example: Please insert your high-level architecture design diagram representing your "current" configuration. This diagram may be different from you "desired" configuration.

M.7.2 Hardware Requirements

M.7.2.1 Hardware Configuration

Instructions: In section it is required that you list in detail the hardware configuration, including peripheral equipment, required to support the business application during the disaster recovery period. Ensure that special hardware needs are listed.

Server Name	
Model	
OS Version	
CPU(s)/Mhz	
Storage Requirements: Internal/External	
Network Interface Requirements	

© 2010 by Taylor and Francis Group, LLC

M.7.2.2 Software Requirements and Vendor Support

Instructions: In this section it is required that you list in detail all infrastructure software required for the application during the disaster recovery period, both system software and application software. This includes, but is not limited to, database, operating system and level, and all requisite software requirements. Give vendor contact information for any application that is based on a vendor software package.

Give details of any contractual limitations to the use of vendor software in a disaster. List key vendor contact persons in the "**Support Notification List**" section of this document (Section M.10). Document instructions covering licensing requirements or issues, override requirements, and location of special passwords that may be required during recovery.

Software Table

List Software Required				
Application Software	*Version/ Release*	*Vendor*	*Platform*	*Contractual Limitations*

M.7.2.3 Special Software Configuration Instructions

In this section or in the Software Table above, specialized installation instructions must be provided.

M.7.3 Business-Critical Data

This section requires that information is provided on *all* business-critical data files received from and/or sent to other business applications, any special internal files, libraries, control records, etc**. It is irrelevant whether the sending or receiving applications themselves are considered business critical or not.** In other words, if the business application input files or output files have not been identified, backed up, and stored off premise, then the application would be in serious jeopardy of not being recovered in case of a disaster. This is especially true for "same-day" and "noncritical" applications.

© 2010 by Taylor and Francis Group, LLC

Order	In/ Out	Application To Be Recovered	Platform Type	Location of Platform
1.	In			
2.	In			
3.	–	<Application>		
4.	Out			
5.	Out			

M.7.3.1 Workaround If Input Files Are Not Available

Instructions: Indicate steps to be taken if input files are not available.

Enter instructions here.

M.7.3.2 <Application> Impact

This section defines interfacing systems that would be affected if the <Application> is not available within the specified recovery time.

Instructions: Indicate, in the appropriate order, the applications that must be recovered with the application and whether they are predecessor (In) or successor (Out) applications. Include the application in the listing.

Order	In/ Out	Application To Be Recovered	Platform Type	Location of Platform
1.	In			
2.	In			
3.	–	<Application>		
4.	Out			
5.	Out			

© 2010 by Taylor and Francis Group, LLC

M.8 Recovery Information

M.8.1 Redundancies

Please list all system and network redundancy and locations.

Configuration Item	Redundancy	Location

M.8.2 Dependencies

Please list system, infrastructure, service, facility, or interface dependencies (in priority order). This will identify related recovery plans or procedures that will need to be invoked in conjunction with this recovery plan. The related plans can be identified and annotated with this plan.

System	Relationship	Contact

M.9 Recovery Execution

M.9.1 Risk Assessment Worksheet

This section has been included to provoke thoughts to help ensure that all needed recovery procedures have been considered to help assess risk to mitigation.

© 2010 by Taylor and Francis Group, LLC

Risk	*Mitigation*
Loss of internal IT systems/networks	
Loss of external IT systems/networks	
Loss of data	
Unavailability of key technical and support staff	
Failure of service providers, e.g., network providers	
(Insert other possible risk)	
(Insert other possible risk)	
Physical loss of a data center	
Loss of all telecom carrier service	
Data center failure (power or telecommunications)	

M.9.2 Recovery Procedures

Please document detailed recovery procedures that will need to be used to recover the subject IT service. Insert or attach the detailed step-by-step recovery procedure(s), any supporting documentation, user guides, and diagrams that will be used to mitigate the recovery of the IT service. These procedures are expected to correlate to the infrastructure tier levels (1 through 4) RTO and RPO. Consideration should be given to "Day 1" functionality and full recovery functionality. Consideration should also be given to the magnitude of the outage, and whether it is an isolated event or catastrophic. Be sure to document any dependencies needed to achieve recovery.

M.9.2.1 Application Recovery Script Process for Day 1 Recovery

M.9.2.2 Application Recovery Script Process for Full Recovery

Define step-by-step scripts of what is to be done during the different phases of recovery after a disaster when the infrastructure has been recovered. This description may be based on the Disaster Recovery test scripts developed for this application.

© 2010 by Taylor and Francis Group, LLC

In support of the recovery scripts, include how to handle restoration of the Application:

- Escalation procedures
- Organizational and technical procedures when the system is available again, including system verification before production
- Organizational and technical procedures in a backup situation knowing that production will have to come back to normal. Describe the following:
 - Who has to do it
 - Estimated time
 - Resources needed
 - Expected results

M.9.2.3 Requirements for Initial Batch Processing Prior to Restoration

M.9.2.4 Requirements for Initial Batch Processing after Restoration

M.9.2.5 Requirements for Normal Batch Processing after Restoration

M.10 Support Notification List and Responsibilities

M.10.1 Management Notification

The recovery team is composed of TFS and members from the business community. The following is a list of each position and a brief overview of each member's responsibilities: Make changes where applicable.

Management:

Primary	Home/Work/Pager/Cell Numbers: *	Responsibility:
Alternate	Home/Work/Pager/Cell Numbers: *	Responsibility:

© 2010 by Taylor and Francis Group, LLC

Crisis Management:

Primary	Home/Work/Pager/Cell Numbers: *	Responsibility:
Alternate	Home/Work/Pager/Cell Numbers: *	Responsibility:

*Mandatory contact information

M.10.2 IT Technical Support Notification

Information Technology: Application

Primary	Home/Work/Pager/Cell Numbers: *	Responsibility:
Alternate	Home/Work/Pager/Cell Numbers: *	Responsibility:

Information Technology: Hardware

Primary	Home/Work/Pager/Cell Numbers: *	Responsibility:
Alternate	Home/Work/Pager/Cell Numbers: *	Responsibility:

Information Technology: Software

Primary	Home/Work/Pager/Cell Numbers: *	Responsibility:
Alternate	Home/Work/Pager/Cell Numbers: *	Responsibility:

Information Technology: Network

Primary	Home/Work/Pager/Cell Numbers: *	Responsibility:
Alternate	Home/Work/Pager/Cell Numbers: *	Responsibility:

Information Technology: Database

Primary	Home/Work/Pager/Cell Numbers: *	Responsibility:
Alternate	Home/Work/Pager/Cell Numbers: *	Responsibility:

Information Technology: Database

Primary	Home/Work/Pager/Cell Numbers: *	Responsibility:
Alternate	Home/Work/Pager/Cell Numbers: *	Responsibility:

Information Technology: Operations and Production Support

Primary	Home/Work/Pager/Cell Numbers: *	Responsibility:
Alternate	Home/Work/Pager/Cell Numbers: *	Responsibility:

*Mandatory contact information

© 2010 by Taylor and Francis Group, LLC

M.10.3 Vendor Key Contacts

Hardware

Primary	Home/Work/Pager/Cell Numbers: *	Responsibility:
Alternate	Home/Work/Pager/Cell Numbers: *	Responsibility:

Software

Primary	Home/Work/Pager/Cell Numbers: *	Responsibility:
Alternate	Home/Work/Pager/Cell Numbers: *	Responsibility:

Network

Primary	Home/Work/Pager/Cell Numbers: *	Responsibility:
Alternate	Home/Work/Pager/Cell Numbers: *	Responsibility:

*Mandatory contact information

M.10.4 Users Required for Recovery

A list of user actions and roles required for recovery of the system or application.

Item	*Action/ Function*	*When Needed*	*Role or Person*	*Exception Handling*
1.				
2.				
3.				
4.				
5.				

© 2010 by Taylor and Francis Group, LLC

M.10.5 Recovery Checklist Validation

To facilitate the execution of key activities in an expeditious manner, the checklist should include key recovery tasks. The tasks need to correlate to the recovery procedures listed below. **Use this form *or* attach a detailed project plan**.

Task	*Target Completion*	*Actual Completion*

Attachment A: Roles and Responsibilities

	Functional Manager	*Project Team*	*Functional Team*	*Release Mgt*	*ITSCM*
Creation of IT continuity plan	A	R	C	I	C/I
Testing of plan	A	R	R/C		C
Acceptance of plan	A	R	R	C/I	C/I
Continued update	A		R	I	C/I
Plan execution (if needed)	A		R		C

Note: A – Accountable (oversees task to ensure it is completed according to expected standard), R – Responsible (performs action, assignment task), C – Consultative (provides additional information and helps determine, presence required in meetings), I – Informed (provide with updates, no action required)

© 2010 by Taylor and Francis Group, LLC

Attachment B: IT Recovery Planning Deliverables Checklist

The following IT disaster recovery information is required. All items must be provided or addressed.			
YES	NO	N/A	
☐	☐	☐	Provide a detailed recovery process to restore the business application.
☐	☐	☐	Provide a detailed list of *all* business-critical business application databases.
☐	☐	☐	Provide a detailed list of *all* business-critical business application data files.
☐	☐	☐	Provide a detailed list of *all* business-critical business application libraries.
☐	☐	☐	Provide a detailed list of computer hardware requirements.
☐	☐	☐	Provide a detailed list of computer software requirements.
☐	☐	☐	Provide a current and detailed business impact assessment, if known.
☐	☐	☐	Identify and provide a detailed list that defines *all* of the inter- and intradependencies of the business-critical business application.

Attachment C: Application Architecture Diagram

Attach a detailed architecture diagram of the application.

Attachment D: Definitions

For the purpose of this document, an understanding of key concepts is important to be able to differentiate among service interruption, business continuity, and disaster recovery. When a service interruption occurs, the process of service restoration is achieved through the incident lifecycle.

business continuity (BC): The ability of an organization to provide service and support for its customers and to maintain its viability before, during, and after a severe outage event. Note: This is specific to each business unit.

business impact analysis (BIA): A process designed to prioritize business functions by assessing the potential quantitative (financial) and qualitative

© 2010 by Taylor and Francis Group, LLC

(nonfinancial) impact that might result if an organization were to experience a business continuity event.

crisis management (CM): The overall coordination of an organization's response to a crisis, in an effective, timely manner, with the goal of avoiding or minimizing damage to the organization's profitability, reputation, and ability to operate.

crisis management team (CMT): A team consisting of key executives, key role players (i.e., media representative, legal counsel, facilities manager, disaster recovery coordinator, etc.), and the appropriate business owners of critical functions who are responsible for recovery operations during a crisis.

disaster recovery (DR): A series of activities, programs, and processes that focus on the ability to recover IT services to support the recovery of critical business functions in response to physical disasters or significant disruption. These processes are contained in the business continuity plan (BCP). DR can be achieved by restoring IT operations at an alternate location, with alternate equipment or manual methods.

recovery point objective (RPO): From a business perspective, RPO is the maximum amount of data loss the business can sustain during an event. The RPO is the targeted point in time to which systems and data must be recovered after an outage as determined by the business unit. (Example: if the RPO is one hour, backups must be made at least once per hour. If the RPO is five days, backups must be made at intervals of five days or less.)

recovery time objective (RTO): The agreed upon time within which systems, applications, or functions must be recovered after an outage. (Example: the RTO for a payroll system might be as long as two weeks, while the RTO for a sensitive financial process might be effectively zero, requiring that service be restored almost instantly.) RTOs are often used as the basis for the development of recovery strategies, and as a determinant as to whether or not to implement the recovery strategies during a disaster situation.

service continuity management (SCM): Mitigates the impact on the IT services required to support the critical business functions and processes. The loss of critical business functions and processes, such as financial loss, damage to reputation, or regulatory breach, are measured through a business impact analysis (BIA), which determines the minimum critical requirements.

service continuity plan (SCP): Focuses on the IT services required to support the critical business functions and processes. The impact of a loss of a business functions and process, such as financial loss, damage to reputation, or regulatory breach, are measured through a business impact analysis (BIA), which determines the minimum critical requirements.

service interruption: An IT service is interrupted and it is not performing at the agreed upon and designed performance levels. A service interruption may be part of a major business disruption; however, most service interruptions are not.

© 2010 by Taylor and Francis Group, LLC

Appendix N: Post-Implementation Review*

Contents

Revisions 306
Record of Modifications 306
Template Owner 306
Acronyms 307
N.1 Scope Fulfillment 307
N.1.1 Scope Summary 307
N.1.1.1 Objectives and Goals Achieved 307
N.1.1.2 Objectives and Goals Missed 307
N.2 Problem Areas, Impact, and Analysis 307
N.2.1 Demand Planning 308
N.2.1.1 Scope 308
N.2.1.2 Design 308
N.2.1.3 Development 308
N.2.1.4 Test Cycle 308
N.2.1.5 Training 308
N.2.1.6 Implementation 308
N.3 Budget 308
N.3.1 Budget Values 308
N.3.2 Budget Deficit 308
N.4 Quality Assurance Statistics 309

* [Release Name]
Post-implementation Review
Version x.x

© 2010 by Taylor and Francis Group, LLC

N.4.1 Release Totals 309
N.4.2 Implementation Totals 309
N.5 Service Level Agreements 309
N.5.1 Performance 309
N.5.2 Incidents and Problems 309

Revisions

Record of Modifications

Below is a record of all modifications made to this template.

Version	*Date*	*Name*	*Details*

Template Owner

The individual named below is the owner of this template. No modifications can be made to the template without going through this individual. All modifications made to the template will be recorded by the template owner in the "Record of Past Modifications" section above.

<Insert name of owner>	
Job Title:	
Company:	
Dept:	
Phone:	
Mail Drop:	
Location:	
E-mail:	

© 2010 by Taylor and Francis Group, LLC

Acronyms

Acronyms	*Description*
CI	configuration item
MTP	master training plan
OR	Operational Readiness
ORT	operational readiness test
SLA	service level agreement
SLM	service-level management
SLO	service level objective

N.1 Scope Fulfillment

Did the release meet the business requirements originally scoped for the release? If not, please explain.

N.1.1 Scope Summary

[Enter text here.]

N.1.1.1 Objectives and Goals Achieved

[Enter text here.]

N.1.1.2 Objectives and Goals Missed

[Enter text here.]

N.2 Problem Areas, Impact, and Analysis

What were the major problems faced in this release in each phase of the release lifecycle? What was the impact to the release schedule and/or the project as a whole? Please provide suggestions as to how these problems can be avoided in future releases.

Some examples might include scope creep, resources changes, build issues, regression testing issues, bugs, vendor issues, training, turnover issues, and so on.

© 2010 by Taylor and Francis Group, LLC

N.2.1 Demand Planning

[Enter text here.]

N.2.1.1 Scope

[Enter text here.]

N.2.1.2 Design

[Enter text here.]

N.2.1.3 Development

[Enter text here.]

N.2.1.4 Test Cycle

[Enter text here.]

N.2.1.5 Training

[Enter text here.]

N.2.1.6 Implementation

[Enter text here.]

N.3 Budget

N.3.1 Budget Values

Enter the dollar amount for each category.

Original Budget Allotment:	$
Final Accrued Budget:	$
Deficit or Surplus:	$

N.3.2 Budget Deficit

If the release went over budget, please explain the underlying causes.

[Enter text here.]

© 2010 by Taylor and Francis Group, LLC

N.4 Quality Assurance Statistics

N.4.1 Release Totals

Enter the total number of open and closed bugs for the entire project for each severity rating, the number of builds or iterations completed during the project's test cycle, and when the back-out plan was tested.

Severity 1 Bugs: [Enter number]
Severity 2 Bugs: [Enter number]
Severity 3 Bugs: [Enter number]
Severity 4 Bugs: [Enter number]
Severity 5 Bugs: [Enter number]
Number of Builds Completed during QA Test Cycle: [Enter number]
Back-Out or Recovery Plan Tested: [Enter number]

N.4.2 Implementation Totals

Severity 1 Bugs: [Enter number]
Severity 2 Bugs: [Enter number]
Severity 3 Bugs: [Enter number]
Severity 4 Bugs: [Enter number]
Severity 5 Bugs: [Enter number]
Number of Workarounds: [Enter number]
Number of Known Issues: [Enter number]

N.5 Service Level Agreements

Since the release, has the product or system performed as expected? Have target SLAs been maintained since the implementation? If no, explain why.

N.5.1 Performance

Expected Performance vs. Actual Performance

[Enter text here.]

N.5.2 Incidents and Problems

List and describe the number of incidents and problems that have been experienced due to the release. Include clarify ticket numbers.

[Enter text here.]

© 2010 by Taylor and Francis Group, LLC